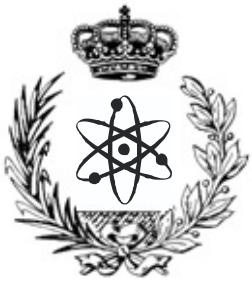


Titulación:

MÁSTER EN INGENIERÍA BIOMÉDICA

Título del proyecto:

COMPARATIVA DE ALGORITMOS DE VISIÓN MONOCULAR PARA LA ESTIMACIÓN DE LA POSICIÓN DE LA CABEZA

José Javier Bengoechea Irañeta

Director: Rafael Cabeza Laguna

Codirector: Mikel Ariz Galilea

Pamplona, 14 de febrero de 2014

Agradecimientos

No podía empezar esta parte con alguien que no fuera Rafa, tutor del trabajo. Gracias por toda esa dedicación durante estos meses. Después de tantas horas y alguna que otra idea de bombero ha salido un trabajo muy completo.

Gracias también a Mikel por todas esas horas y por engañar a gente para que sean nuestros conejillos de indias.

Gracias a Arantxa, en este trabajo hemos coincidido menos, pero gracias por todo.

Cristian y David, se está bien en el laboratorio con vosotros por ahí. Bueno, sin vosotros también se está bien. Pero con vosotros mejor. ☺

Gracias a los compañeros del máster por esos buenos ratos en clase, especialmente a Rebeca, después de tantos trabajos, tanto tiempo codo con codo y esos meses en el laboratorio, me da pena que esto se termine.

1 – Introducción

En los últimos años la instrumentación médica está sufriendo un cambio significativo en cuanto a invasividad y comodidad para el paciente. Destaca por ejemplo la cirugía laparoscópica, cuya función es realizar algún tipo de cirugía en el interior del paciente realizando pequeñas incisiones por las que el cirujano opera el instrumental. Para ello se ayuda de, entre otras cosas, un sistema de imagen que permite ver el interior del paciente sin necesidad de recurrir a grandes incisiones que suponen un postoperatorio más largo.

Es también interesante el avance en radioterapia externa. Las últimas investigaciones se dedican a desarrollar sistemas de localización del tumor en (casi) tiempo real, con el fin de ajustar el tratamiento para irradiar células tumorales sin dañar el tejido adyacente. Esto se lleva a cabo mediante sistemas de imagen radiológica.

Otras aplicaciones pensadas no para el ámbito quirúrgico sino para el ámbito asistencial en el hogar, como los sistemas de estimación de la mirada, buscan suplir las limitaciones de personas con movilidad reducida en tareas cotidianas como el manejo de un ordenador.

Como se puede observar, los avances en la tecnología están íntimamente ligados a los sistemas de imagen. Siguiendo la misma filosofía, se empiezan a utilizar sistemas de imagen en dispositivos médicos en los que existe movimiento del paciente, bien sea para evitar artefactos de movimiento y aumentar la calidad de imágenes radiológicas (TAC, MR, PET), bien sea para reducir el margen de error de un tratamiento (radioterapia en pulmones en movimiento).

Una de las líneas de investigación del grupo de ingeniería biomédica, dentro del proyecto “Universalización de las interfaces de ordenador basadas en seguimiento de la mirada” se centra en desarrollar un sistema de estimación de la posición de la cabeza con el objetivo principal de integrarlo en un sistema de estimación de la mirada, si bien es una aplicación que se puede aplicar en otros campos, como en la reducción de artefactos de movimiento en imágenes radiológicas craneales.

Este proyecto busca estudiar una serie de algoritmos de estimación de la posición de la cabeza, y analizar su viabilidad como solución a problemas médicos actuales. Además, se pretende crear una base de datos de imágenes de movimientos de cabeza de alta calidad con la que probar los algoritmos.

2 – Objetivos

El primer objetivo de este trabajo es la realización de una base de datos de posiciones de cabeza en 3D, consistente en varias personas realizando diferentes movimientos de cabeza, utilizando una cámara web de visión monocular y el sensor trakSTAR 3D Guidance para conocer la posición y orientación de la cabeza de los usuarios.

La información relativa a la posición de la cabeza debe estar en el sistema de referencia (desde el punto de vista) de la cámara. La información del sensor viene dada en el sistema de coordenadas del transmisor, de manera que es preciso calcular la relación entre el sistema de coordenadas de la cámara y el sistema de coordenadas del transmisor, un proceso denominado calibración, de manera que sea posible relacionar dos sistemas de coordenadas a priori independientes. Partiendo del proyecto "Desarrollo de una base de datos de posiciones 3D de la cabeza empleando el sensor trakSTAR 3D Guidance Studio" de Rebeca Echeverría [1][2], en este trabajo se busca simplificar el proceso de calibración, reducir el tiempo necesario para tal tarea y aumentar la precisión en los resultados.

Las imágenes de la base de datos contienen una serie de marcas que definen las diferentes estructuras faciales. Dada la inviabilidad de marcar las imágenes de forma manual, se implementa un sistema de marcado automático, utilizando el transmisor y dos sensores y unas piezas diseñadas para tal efecto.

El segundo objetivo de este trabajo es evaluar diferentes algoritmos de estimación de la posición de la cabeza, analizando parámetros como la precisión en los resultados, la estabilidad o el tiempo de procesado.

Los métodos propuestos constan de dos etapas. La primera consiste en detectar una serie de puntos faciales, para lo que se utilizan los algoritmos de segmentación ASM y AAM. La segunda etapa consiste en utilizar la información de esos puntos para estimar la posición de la cabeza para lo que se utiliza el algoritmo POSIT. Dado que es necesario un modelo tridimensional de la cabeza para POSIT, y que cada persona tiene una cabeza diferente, se estudia también la viabilidad del uso de modelos genéricos y deformables.

Además se estudian dos sistemas de estimación ya existentes. El primero es FaceAPI, un sistema comercial de SeeingMachines, el segundo es Intraface, un sistema de libre acceso desarrollado por HumanSensing.

Finalmente, se estudia el resultado de la estimación de la posición de la cabeza utilizando las marcas automáticas obtenidas para la base de datos, con el fin de estimar qué parte del error se debe a la segmentación de las estructuras faciales y qué error se debe a POSIT.

3 – Aplicaciones

La estimación de la posición de la cabeza en el ámbito médico se está utilizando en estudios o tratamientos de larga duración en los que el movimiento de la cabeza del paciente puede provocar errores importantes, bien sea en el estudio médico que se está llevando a cabo o bien en el tratamiento que recibe el paciente. Por ejemplo, las pruebas de imagen por medicina nuclear (PET y SPECT) son pruebas de muy larga duración temporal y el movimiento del paciente puede causar una notable pérdida de calidad en la imagen [3][4]. Otro ejemplo interesante es la estimulación magnética transcraneal [5], en la que se estimula una región del cerebro muy concreta, y un error en la localización de la zona puede tener consecuencias graves. Por otro lado, se está avanzando notablemente en los sistemas de *eye-tracking* y *gaze estimation* para discapacitados. Las técnicas más recientes utilizan un estimador de la posición de la cabeza para ajustar la posición relativa de los ojos frente a la cámara [6].

3.1 - Medicina nuclear

La Tomografía por Emisión de Positrones es una técnica de diagnóstico por imagen que mide la actividad metabólica del cuerpo humano. Al igual que otras técnicas diagnósticas en medicina nuclear como el SPECT, el PET se basa en detectar y analizar la distribución de un radiofármaco en el interior del paciente.

En los órganos en los que se localiza el radiofármaco se producen aniquilaciones positrón-electrón que dan como resultado fotones gamma. Los detectores de un tomógrafo PET, situados formando un anillo alrededor del paciente, detectan los fotones generados en cada aniquilación. Estos fotones se convierten en señales eléctricas, y tras un proceso de filtrado y reconstrucción se obtiene la imagen.

La duración del PET depende del órgano de estudio, pero generalmente se estima una duración de entre 30 y 60 minutos. Durante ese tiempo el detector está constantemente captando fotones. Un factor crítico en la calidad de la imagen es el movimiento del paciente durante la prueba. Si el órgano emisor de fotones varía su posición, el sistema lo identifica como un emisor válido, y la imagen resultante contiene gran cantidad de ruido.

Para corregir este error se están estudiando técnicas de estimación del movimiento para la corrección de la señal durante la reconstrucción de la imagen [3][4]. En el caso de PET craneal, se utiliza un sistema con marcadores fiduciales externos y un sistema de visión estéreo, que detecta la posición y orientación de la cabeza y corrige la detección (Fig. 3.1).

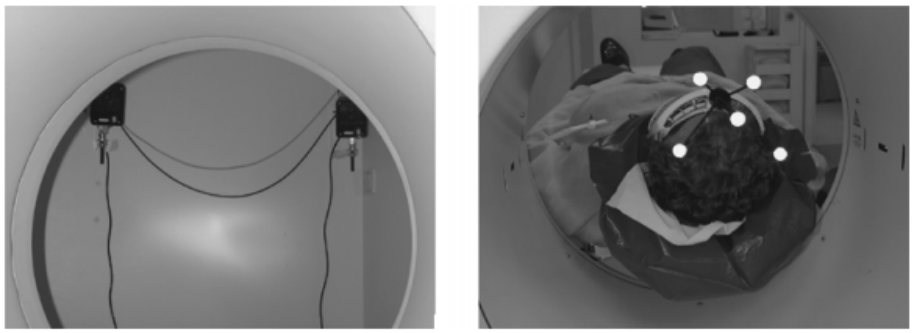


Fig. 3.1: sistema de estimación de movimiento basado en imagen en PET

La calidad de la imagen obtenida mediante este sistema de compensación de movimiento es notable. En la figura 3.2 se muestran 3 imágenes. La imagen de la izquierda muestra un PET del cerebro sin movimiento durante la prueba. La imagen del centro muestra un PET del cerebro en el que el paciente se ha movido durante la prueba. La imagen de la derecha muestra el mismo PET de la segunda imagen tras la corrección de los movimientos del paciente.

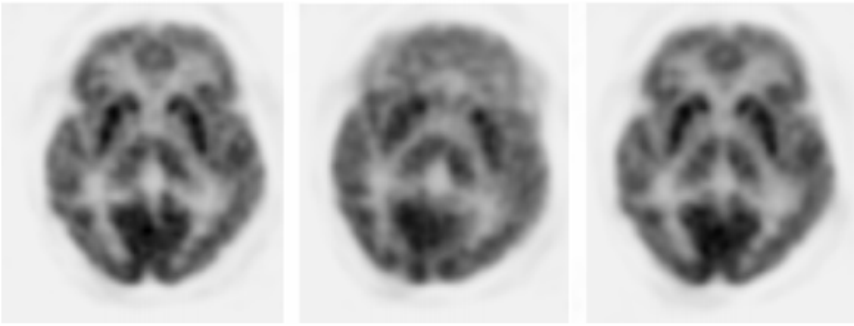


Fig. 3.2: calidad de imagen en PET sin y con corrección de movimiento

3.2 - Estimulación magnética transcraneal (EMT)

La estimulación magnética transcraneal (EMT) es una técnica de estimulación cerebral indolora y no invasiva, mediante la inducción de corrientes eléctricas en el cerebro, utilizada para el estudio y tratamiento de algunas enfermedades neurológicas y psicológicas. La EMT suele utilizarse en el estudio de vías motoras centrales, para el estudio de la excitabilidad cortical y en el mapeo de funciones cerebrales corticales.

Debido a su alta resolución espacial y temporal, y por ser capaz de activar o interferir con funciones cerebrales, permite establecer relaciones causales entre la actividad cerebral y el comportamiento del paciente, y no sólo correlaciones como ocurre con otras técnicas neurofisiológicas o de neuroimagen.

El componente principal en la EMT es el estimulador magnético. Consiste básicamente en una bobina de hilo de cobre por la que fluye un pulso de corriente que genera un campo magnético. La variación en el campo magnético de la bobina induce una corriente secundaria en el cerebro del paciente.

En la EMT actual, el médico/ físico sitúa la bobina manualmente en la cabeza y la mantiene ahí durante el tratamiento de la forma más precisa que le es posible. Dado que un tratamiento puede durar entre 20 y 30 minutos, y que se necesita una precisión inferior a unos pocos milímetros, están surgiendo sistemas robotizados que posicionan la bobina con precisión y corrigen la posición en función de los movimientos del paciente.

Para la compensación del movimiento se utiliza un sistema de *tracking* formado por una cámara estéreo y unos marcadores colocados en la cabeza del paciente [5]. Cada vez que el paciente realiza un movimiento de la cabeza, el sistema detecta la nueva posición con un error inferior a 0.25 mm y ajusta el brazo robótico para situar la bobina en el lugar correcto.

La figura 3.3 muestra un dispositivo de EMT compuesto del estimulador, brazo robótico y sistema de marcadores para *tracking* y corrección de movimiento.

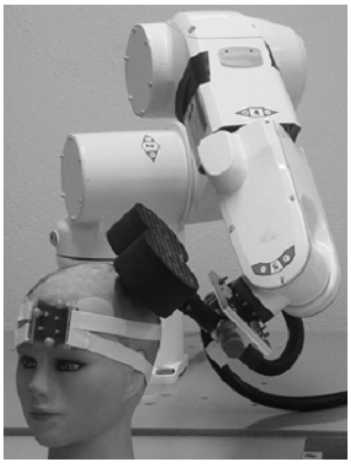


Fig. 3.3: sistema de EMT robotizado

3.3 - *Eye Tracking y gaze estimation*

Muy conocidas son las técnicas de estimación de la mirada utilizando la información de la posición de la cabeza y la de los ojos frente a ésta. La principal aplicación de estos sistemas en el ámbito asistencial es la de controlar un dispositivo, generalmente el cursor de un ordenador, utilizando únicamente la mirada, muy interesante para personas discapacitadas.

Una aplicación menos conocida pero muy curiosa es la detección y cuantificación del estrabismo [6]. El estrabismo consiste en una desviación del alineamiento de un ojo con respecto al otro. Este problema impide fijar la mirada de los dos ojos en un mismo punto espacial, lo que provoca una visión binocular defectuosa, afectando a la estimación de la profundidad.

Existe un sistema para la detección del estrabismo infantil que consiste en utilizar un conjunto de cámaras para estimar la dirección de visión de cada ojo (Fig. 3.4). Este sistema detecta la posición de la cabeza para corregir las estimaciones y además se adapta a los movimientos de cabeza del paciente, que siendo generalmente niños, esos movimientos son muy comunes. De esta manera el sistema es muy robusto y proporciona unos resultados muy precisos.

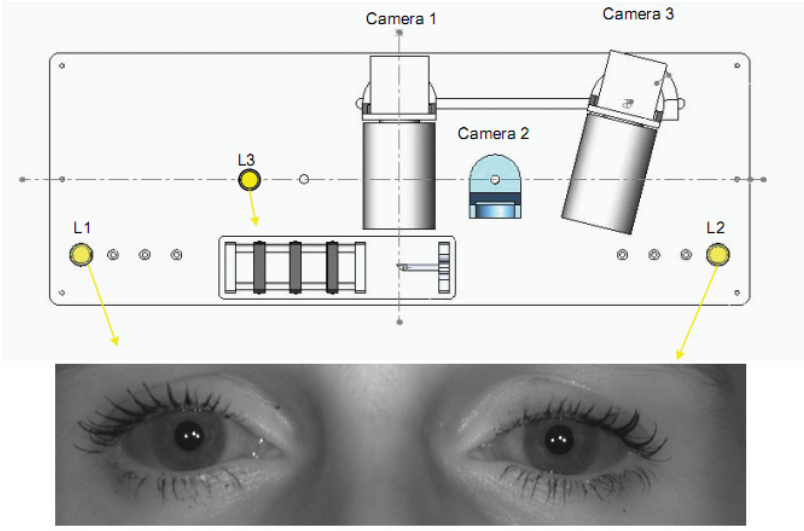


Fig. 3.4: sistema de detección de estrabismo

4 – Materiales utilizados

4.1 - Cámara web Logitech HD Pro C920

La cámara web utilizada es el modelo Logitech HD Pro C920 (Fig. 4.1), permite grabaciones a una resolución de 1280x720 a una frecuencia de adquisición de 30 *frames* por segundo, ideal para la grabación de las imágenes con alta calidad. La resolución real (no interpolada) máxima es de 1920x1080, resolución utilizada para la calibración del sistema.

Fig. 4.1: cámara web Logitech HD Pro C920

4.2 - 3D Guidance TrakSTAR de Ascension Technology Corporation

El dispositivo trakSTAR es un sistema de *tracking* magnético compuesto por un transmisor y varios sensores. Permite conocer la posición y orientación de cada sensor conectado al dispositivo respecto del sistema de coordenadas del transmisor.

Se compone de 3 elementos principales:
- Unidad electrónica: procesa la información de los sensores y controla y alimenta el transmisor (Fig. 4.2 a).
- Transmisor: emisor de campos magnéticos (Fig. 4.2 b).
- Sensores: a partir de la señal del transmisor estiman su posición y orientación y envían la información a la unidad electrónica (Fig. 4.2 c).

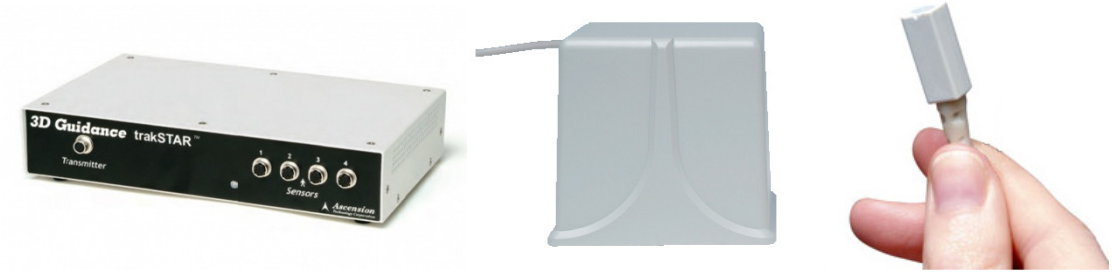


Fig. 4.2: a) unidad electrónica b) transmisor c) sensor

4.3 - Modelos de cabeza

Se han utilizado dos modelos de cabezas humanas, uno de ellos con apariencia humana para las pruebas iniciales (Fig. 4.3 a) y otro sin apariencia humana al que se le ha colocado pegatinas en unos puntos específicos para evaluación de errores (Fig. 4.3 b).

Fig. 4.3: a) modelo real b) modelo irreal

4.4 - Sistema de iluminación

La grabación de los videos se ha realizado en condiciones de iluminación controladas. Para ello se ha prescindido de la iluminación natural y se ha utilizado únicamente iluminación de estudio.

Se han utilizado 3 focos con sus correspondientes *dimmers* (reguladores de intensidad), un reflector y un difusor, con el objetivo de generar luz difusa sin sombras y una buena iluminación ambiental para dar un mayor realismo a las imágenes.

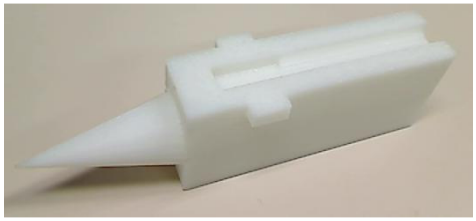
4.5 - Piezas para el manejo de los sensores

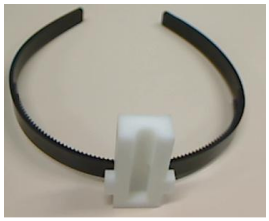
Para manejar los sensores de manera precisa se han diseñado y construido 3 piezas de plástico rígido.

La primera de ellas es un útil con un hueco para la inserción de un sensor y un extremo afilado para marcar puntos en el espacio (Fig. 4.4 a).

La segunda es la pieza utilizada durante las grabaciones para colocar el sensor en la cabeza del sujeto. Se acopla a una diadema de manera firme, de modo que durante todo el proceso de grabación el sensor permanece solidario con la cabeza (Fig. 4.4 b).

La tercera se utiliza para pruebas con el modelo de cabeza humano, está pegada a la cabeza de manera que el sensor permanece en el mismo punto en todo momento (Fig. 4.4 c).





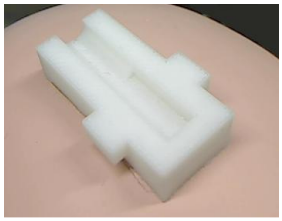


Fig. 4.4: a) útil de marcado b) pieza para la cabeza c) pieza para el modelo

5 – Descripción de la base de datos

5.1 - Videos

La base de datos consiste en una serie de videos de diferentes personas realizando movimientos de cabeza. Se ha tomado a 10 personas y con cada una de ellas se ha grabado 12 videos. De esos 12 videos, 6 siguen un patrón de movimientos predefinido (traslaciones y rotaciones puras) y los 6 restantes consisten en movimientos libres, a discreción del sujeto, combinando movimiento en los 6 grados de libertad.

El sistema de coordenadas de la cámara es el siguiente (Fig. 5.1 a):
- Eje X: horizontal, positivo hacia la derecha.
- Eje Y: vertical, positivo hacia abajo.
- Eje Z: profundidad, positivo alejándose.

Los giros de la cabeza en los 3 ejes se denominan *roll*, *yaw* y *pitch* (Fig. 5.1 b):
- *Roll*: consiste en girar la cabeza intentando tocar los hombros con las orejas.
- *Yaw*: consiste en girar la cabeza hacia los lados, como en el gesto de negación.
- *Pitch*: consiste en girar la cabeza hacia arriba y hacia abajo, como en el gesto de afirmación.

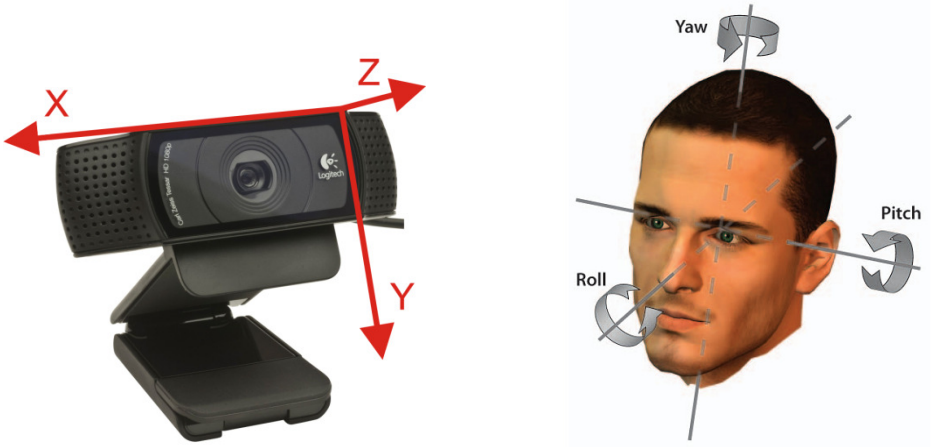


Fig. 5.1: a) coordenadas de la cámara b) giros de la cabeza

Los videos de cada usuario siguen un orden preestablecido, de modo que en cualquiera de los 10 usuarios el movimiento del sujeto en cada video es el siguiente:

Video 1:	Traslación pura en el eje X, movimiento de izquierda a derecha
Video 2:	Traslación pura en el eje Y, movimiento de arriba a abajo
Video 3:	Traslación pura en el eje Z, movimiento de acercarse y alejarse
Video 4:	Rotación pura de *Roll*
Video 5:	Rotación pura de *Yaw*
Video 6:	Rotación pura de *Pitch*
Videos 7-12:	Movimiento libre como combinaciones de los 6 anteriores

Los videos tienen una duración de 10 segundos con una frecuencia de adquisición de 30 *frames* por segundo, lo que hace un total de 300 *frames* por video. La resolución empleada es 1280 x 720 pixeles, con una relación de aspecto de 16:9.

Para la grabación se ha empleado iluminación de estudio, para obtener imágenes de gran calidad sin presencia de sombras.

La base de datos incluye información sobre la posición y rotación de la cabeza en cada uno de los frames de cada video, en el sistema de referencia de la cámara. Así es posible estudiar la calidad de diferentes algoritmos de estimación de la posición al comparar los resultados con la posición y orientación real del usuario, lo que se denomina *ground truth*.

5.2 - Marcado de las imágenes

Una base de datos de imágenes de calidad para *Head Pose Estimation* debe contener, además de las imágenes y la información referente a la posición de la cabeza, una serie de marcas que identifiquen puntos característicos de la cara, denominadas *landmarks*. Esas marcas se utilizan para entrenamiento y evaluación de algoritmos. Por ejemplo, son comunes los landmarks en los *corners* (esquinas) y parte superior e inferior de los ojos, *corners* de la boca o siguiendo la línea de la mandíbula. En esta base de datos se proponen 54 puntos que caracterizan las regiones más importantes de la cara, de esta manera (Fig. 5.2):

Ceja derecha:	5	siguiendo la línea superior.
Ceja izquierda:	5	siguiendo la línea superior.
Ojo derecho:	8	siguiendo la línea de los párpados.
Ojo izquierdo:	8	siguiendo la línea de los párpados.
Nariz:	11	siguiendo el perfil.
Boca:	8	siguiendo la línea de los labios.
Mandíbula:	9	siguiendo el perfil.

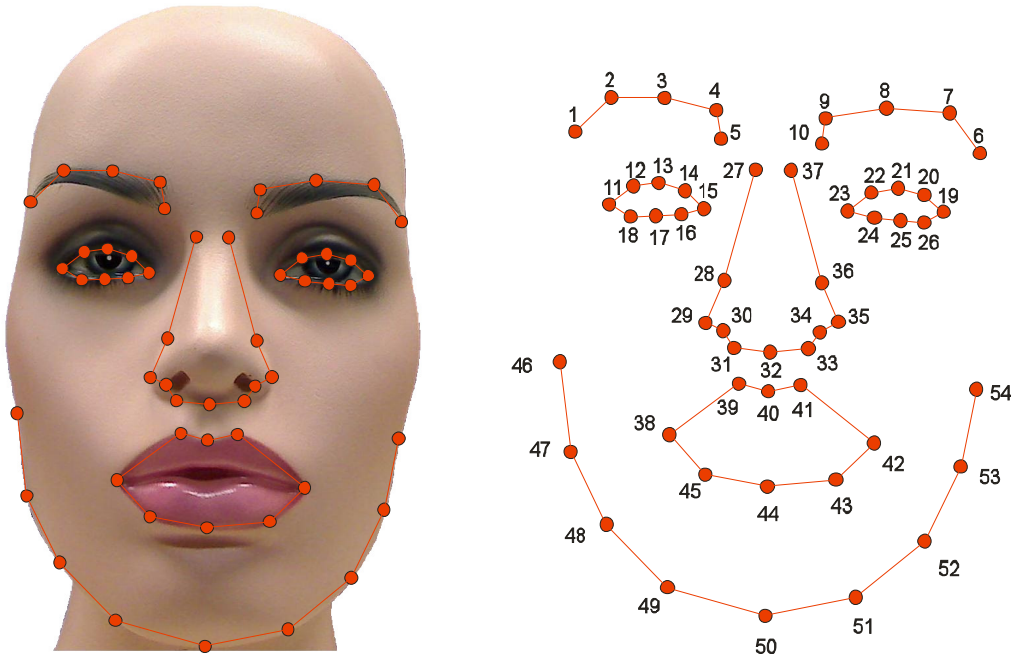


Fig. 5.2: guía del marcado de los 54 puntos faciales

5.3 - Formato de los datos de posición y orientación

La información de la posición y orientación de la cabeza se almacena en un archivo de texto para cada video. El nombre de cada archivo identifica el usuario y el video del que contiene la información. Por ejemplo, el archivo con los datos del video 3 del usuario 1 se llama user_01_video_03_sensor.txt.

Cada archivo consta de 300 filas, una por cada *frame* de vídeo, y 6 columnas, que contienen la posición (en mm) y orientación (en grados) de este modo:

Traslación X (mm) Traslación Y (mm) Traslación Z (mm) *Roll* (º) *Yaw* (º) *Pitch* (º)

Por ejemplo, la posición y orientación de los 4 primeros *frames* de un video sería la siguiente:

```
-24.4867    -141.9389    596.3162    -2.5188    -1.1480    1.3979
-24.4867    -141.9389    596.3162    -2.5417    -1.1503    1.3977
-24.4896    -142.0506    596.3243    -2.5516    -1.1532    1.3974
-24.6006    -142.0478    596.3230    -2.5716    -1.1532    1.3974
```

5.4 - Formato de las marcas

Las marcas se almacenan en un archivo de texto diferente al anterior. En el caso del video 3 del usuario 1, las marcas se almacenan en el archivo user_01_video_03_face_image.txt. Cada archivo contiene 300 filas de datos, una por cada *frame*, y cada fila contiene 108 números, las coordenadas de imagen x e y de cada uno de los 54 *landmarks*. Las marcas siguen el orden de la de la figura X. Cada fila contiene los datos siguiendo este orden:

x_1 y_1 x_2 y_2 x_3 y_3 … x_{54} y_{54}

6 – Calibración del sistema

No es el objetivo de este trabajo explicar detalladamente los fundamentos matemáticos de la calibración, dado que este tema se explica de manera excelente en el proyecto [1]. En este trabajo nos limitaremos a explicar la idea básica del método.

Sí que es el objetivo de este trabajo mejorar el proceso de calibración. Durante el tiempo transcurrido entre el trabajo de [1] y este proyecto han evolucionado los algoritmos de calibración y se considera que es recomendable actualizar las funciones utilizadas anteriormente con el objetivo de obtener una calibración significativamente más precisa y con una duración significativamente inferior. No obstante, lo referente a los cálculos matemáticos de [1] permanece igual; las mejoras se han realizado a nivel software.

6.1 - Diferentes sistemas de referencia

El transmisor tiene un determinado sistema de referencia, con su origen y sus 3 ejes ortogonales, y toda la información que proporciona sobre la posición y orientación del sensor viene dada en ese sistema de referencia (Fig. 6.1 a). Nuestro deseo es obtener la información del sensor en el sistema de referencia de la cámara (Fig. 6.1 b), para que exista una concordancia entre las imágenes grabadas y las posiciones calculadas. El problema radica en que los dos sistemas de referencia son totalmente independientes, y no existe una relación directa entre ambos. Para estimar esa relación se recurre a un proceso de calibración, cuyo fin último es calcular las dos matrices de transformación de sistema de referencia. Una vez calculadas esas matrices, el cambio de sistemas de referencia se limita a simples productos de matrices.

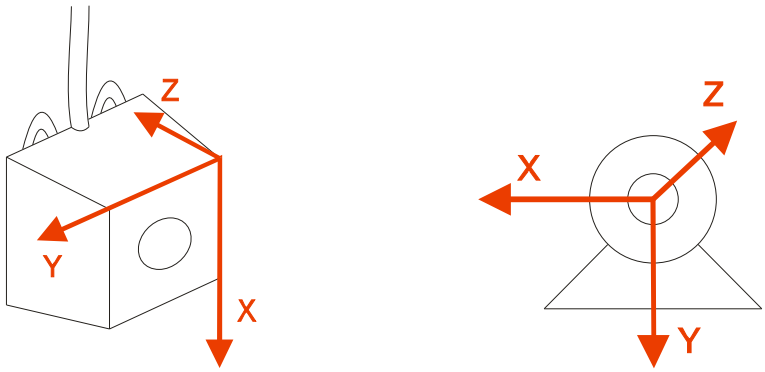


Fig. 6.1: a) sistema de coordenadas del transmisor b) sistema de coordenadas de la cámara

6.2 - Método de calibración

La idea básica de la calibración es que no se puede establecer un camino directo entre la cámara y el transmisor, pero se pueden utilizar otros elementos para definir un camino alternativo. Para ello se utiliza un damero y un sensor del sistema trakSTAR. Tanto el damero como el sensor tienen su propio sistema de referencia (Fig. 6.2).

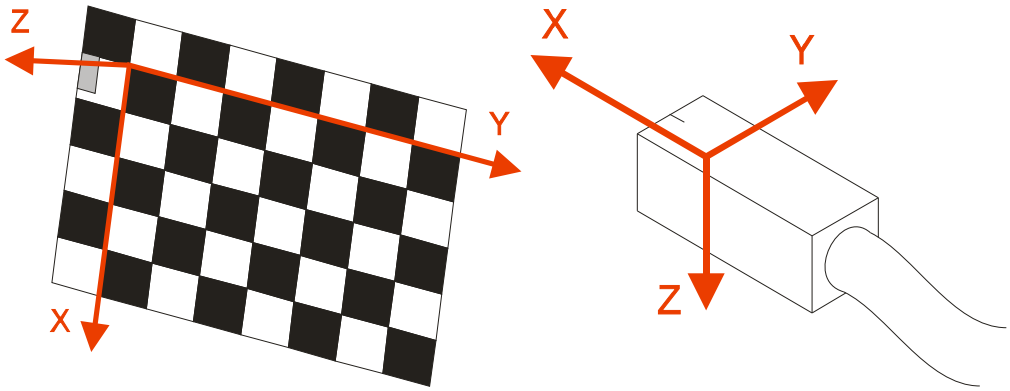


Fig. 6.2: sistemas de referencia del damero y el sensor

Si se toma el damero y se toman varias fotografías del mismo con una cámara en diferentes posiciones y orientaciones, con el software de calibración es posible estimar los parámetros intrínsecos y extrínsecos de la cámara.

Los parámetros intrínsecos son aquellos que caracterizan a la cámara en base a su fabricación, como la distancia focal, el punto principal (origen) y los coeficientes de distorsión de la lente. Los parámetros extrínsecos hacen referencia a la posición y orientación del damero en el espacio. Utilizando los parámetros extrínsecos se puede calcular las matrices de transformación de coordenadas entre la cámara y el damero en cada una de las imágenes, M23 y M32 (Fig. 6.3).

Si se coloca el sensor en el damero en una posición y orientación determinada, y se conoce los sistemas de referencia de ambos (origen y ejes), se puede construir las matrices de transformación de sistema de referencia entre ambos, M12 y M21.

Si además se capturan datos sobre la posición del sensor con respecto al transmisor para cada fotografía del damero, se puede construir las matrices de transformación de coordenadas entre el sensor y el transmisor, M01 y M10. Con estas matrices se construye el camino transmisor – sensor – damero – cámara, y se obtiene las matrices que relacionan el transmisor y la cámara, M03 y M30 (Fig. 6.3).

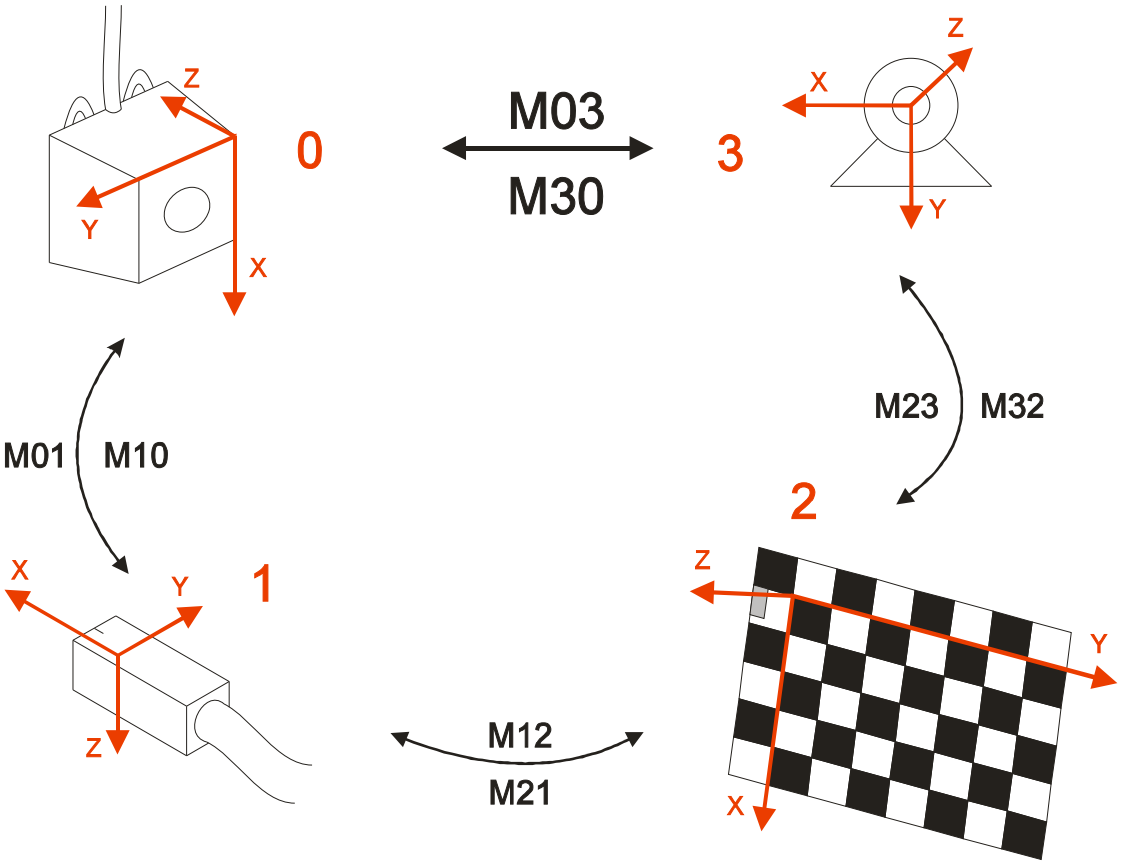


Fig. 6.3: elementos del proceso de calibración

6.3 - Origen de los sensores

Cada sensor tiene su propio sistema de referencia, lo que significa que ambos tienen un origen de coordenadas y unos ejes (Fig. 6.4). El método para la estimación del origen de un sensor está explicado en [1], de manera que no es necesario volver a explicarlo en este trabajo. Se ha calculado el origen de los dos sensores y se concluye que el origen de los dos sensores es:

	Sensor 1	Sensor 2
O_x (mm)	4.9690	4.8189
O_y (mm)	3.8468	3.6544
O_z (mm)	3.5358	3.7631

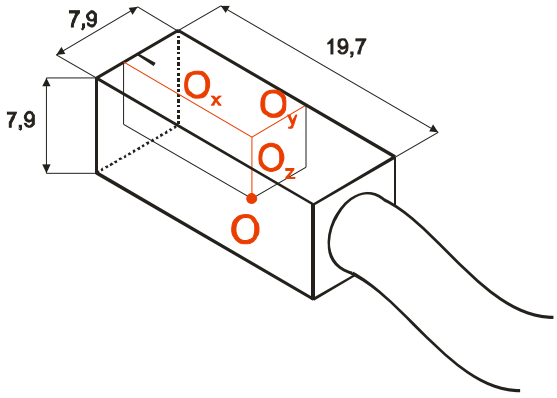


Fig. 6.4: Origen del sensor

6.4 - Adquisición de datos para la calibración

De acuerdo a lo explicado sobre la calibración, es necesario tomar imágenes del damero y a la vez capturar datos con el transmisor sobre el sensor situado en el damero. El método de adquisición de los datos propuesto por [1] es totalmente válido, pero en este trabajo se ha automatizado para poder tomar más datos con menos esfuerzo y en menos tiempo, con la idea de conseguir una calibración mejor y a la vez más rápida.

El método original consiste en tomar las imágenes una a una utilizando el software de la cámara, y los datos del sensor con el software propio del transmisor, almacenarlos en el directorio destino y renombrar los archivos siguiendo un cuidadoso orden.

En este trabajo se ha integrado todo eso en una función, de manera que con pulsar la tecla *enter* se toman todos los datos de manera automática. Además, el programa realiza una búsqueda del damero en la imagen, y si no detecta el damero esperado descarta la imagen y lanza un aviso al usuario para que compruebe la colocación del damero. En ocasiones el software del transmisor falla, así que esta función tiene una componente importante de control de errores, de manera que ante cualquier error de software el programa reacciona y lo corrige, y el usuario no tiene que intervenir.

Utilizando el método de calibración antiguo, se estima que tomar 20 imágenes para calibración con sus correspondientes datos del transmisor puede costar en torno a 40 minutos. Con la nueva función de captura de datos es posible tomar 50 imágenes con sus datos del transmisor en 30 minutos; las 50 imágenes son totalmente válidas para la calibración y la comodidad del usuario es bastante mayor (Fig. 6.5).

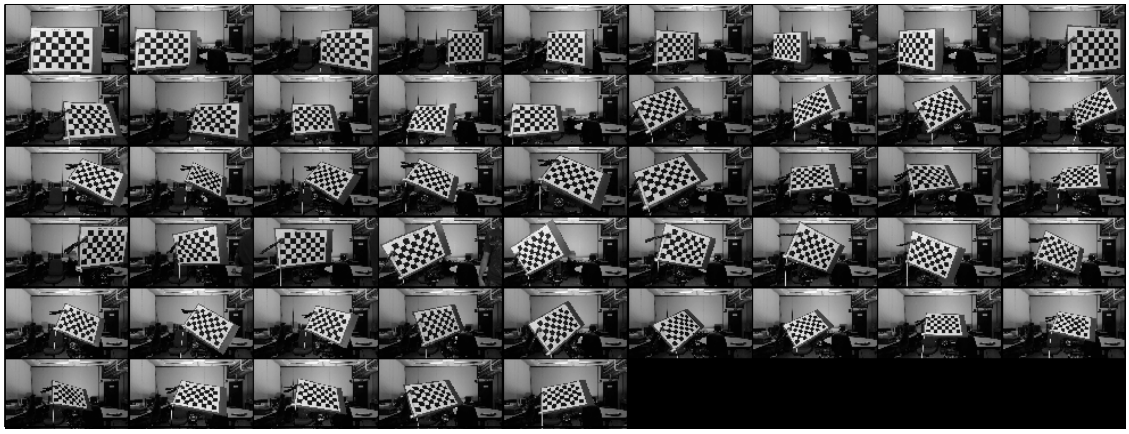


Fig. 6.5: imágenes de calibración

6.5 - Calibración

La calibración consiste en calcular las matrices de transformación de sistemas de coordenadas entre los diferentes elementos de la cadena (transmisor, sensor, damero y cámara), para obtener finalmente las matrices M03 y M30.

A partir de la posición y orientación del sensor con respecto al transmisor, obtener las matrices M01 y M10 es sencillo (véase [1] para más detalles).

Dado que se conoce la posición del sensor en el damero, y se conoce los orígenes y ejes de los sistemas de referencia de ambos elementos, la construcción de las matrices M12 y M21 también es inmediata.

El cálculo de las matrices que relacionan la cámara y el damero en las imágenes de calibración no es sencillo, y para ello en un principio se utilizaba el *toolbox* "*Camera Calibration Toolbox for Matlab*" de Jean-Yves [7], que a pesar de sus limitaciones era el único software existente para esta tarea. La función de calibración proporciona la posición y orientación del damero con respecto a la cámara, con lo que se construyen las matrices M23 y M32. Este *toolbox* tiene una limitación importante, y es que la función de calibración tiene una gran componente manual. Para cada una de las imágenes de calibración hay que indicarle a la función dónde están las 4 esquinas del damero. Esto supone un tiempo considerable y es propenso a cometer errores.

Para solucionar esta limitación se ha utilizado el reciente *toolbox* de calibración de Matlab 2013b, una versión mejorada del *toolbox* anterior que detecta automáticamente el damero en la imagen, de manera que el usuario no tiene más que ejecutar la función de calibración y el proceso se realiza automáticamente sin ningún tipo de marcado manual (Fig. 6.6).

De esta manera, con una actualización de las funciones de calibración se ha conseguido una calibración automática, capaz de emplear más imágenes y con ello obtener unos resultados más precisos en menos tiempo.

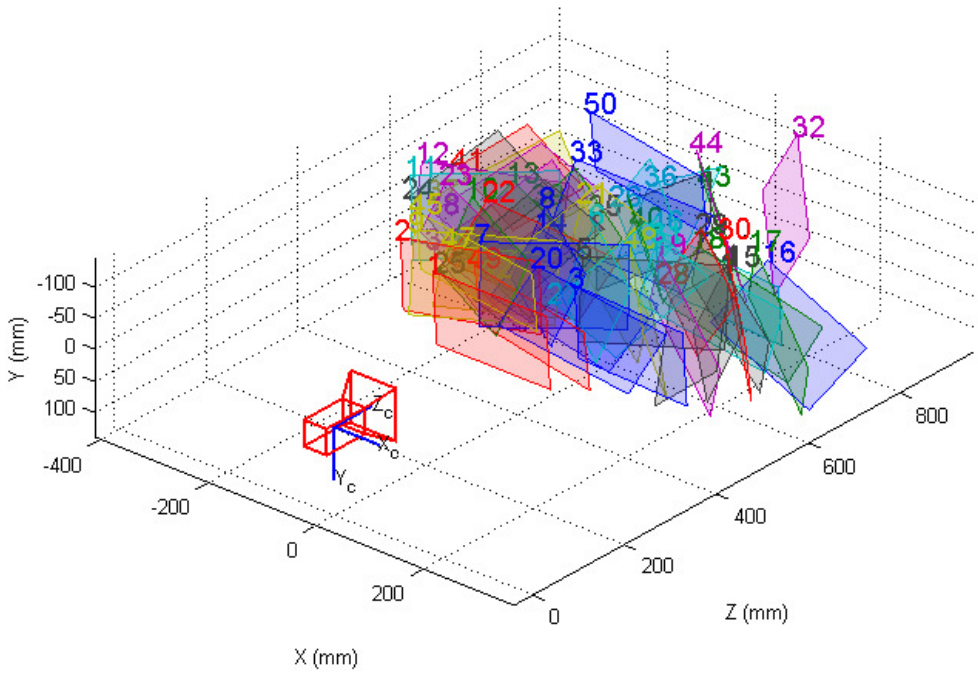


Fig. 6.6: imágenes de calibración

6.6 - Optimización de la calibración

Una vez terminada la calibración se obtienen las matrices M03 y M30. El problema de la calibración es que las matrices resultantes son fruto de una concatenación de etapas, y cada una de ellas es una fuente de error en mayor o menor medida. Para minimizar los errores se ha optimizado cada una de esas etapas.

Error en la medición del sensor (M01 y M10): la calidad de la medida del sensor decae con la distancia al transmisor. Para reducir ese error se ha modificado la disposición del sistema de grabación de manera que el transmisor se sitúa sobre el sujeto fuera del campo de visión de la cámara, y se han retirado todos los objetos metálicos cercanos que pueden interferir con el transmisor.

Error en la estimación de los parámetros de la cámara (M23 y M32): se ha observado que la detección automática del damero es sensible al nivel de iluminación, de manera que una iluminación demasiado alta o baja aumenta el error de calibración. Para reducir ese error se ha optado por tomar las imágenes de calibración en condiciones de iluminación similares a las de la grabación.

Error en la colocación del sensor en el damero (M12 y M21): esta etapa es sin duda la más crítica. En la calibración se asume que el sensor se ha colocado en una región específica del damero, y con una orientación determinada, y se han construido las matrices M12 y M21 en base a esa posición y orientación. La realidad es que la colocación se realiza de forma manual y nunca se consigue la posición y orientación deseadas exactamente, de manera que las matrices anteriores no son del todo correctas. Para reducir el error en este punto se ha empleado una función de optimización que busca la traslación y orientación reales del sensor en el damero, minimizando el error en la proyección de unos puntos conocidos.

La idea de la optimización es la siguiente. Si se conoce la localización espacial de un conjunto de puntos (los vértices del damero) con respecto al transmisor, se puede proyectar esos puntos en el plano imagen utilizando la matriz M03 y los parámetros intrínsecos de la cámara (Fig. 6.7). Si además se sabe la localización exacta de esos puntos en la imagen (por la detección automática del damero), se puede calcular el error de proyección, como la diferencia entre la proyección de los puntos sobre el plano imagen y su localización exacta en la imagen. Dado que el error de calibración depende de la matriz M03, y ésta a su vez depende de M12, el error de calibración es dependiente de M12, es decir, de la relación sensor-damero. Utilizando una función de optimización se pueden obtener los vectores de traslación y rotación que, al construir la matriz M12, minimicen el error de proyección.

Se ha observado que las diferencias entre la matriz M12 antes y después de la optimización son muy pequeñas, con un error en la traslación inferior a 1 mm y un error promedio en la rotación de 1º. En realidad es un error perfectamente asumible para la estimación de la posición de la cabeza con respecto a la cámara, ya que sólo en el proceso de colocación del sensor en la cabeza se tiene una incertidumbre de 2º aproximadamente. Por eso, los resultados obtenidos en el anterior trabajo [1] sin la optimización son perfectamente válidos. No obstante, en este trabajo se ha implementado un sistema de marcado automático muy sensible a errores en la calibración de esta magnitud. Por ello, si bien para una grabación típica no sería necesario optimizar la calibración, en el momento en el que se use el sistema de marcado automático sí será necesario.

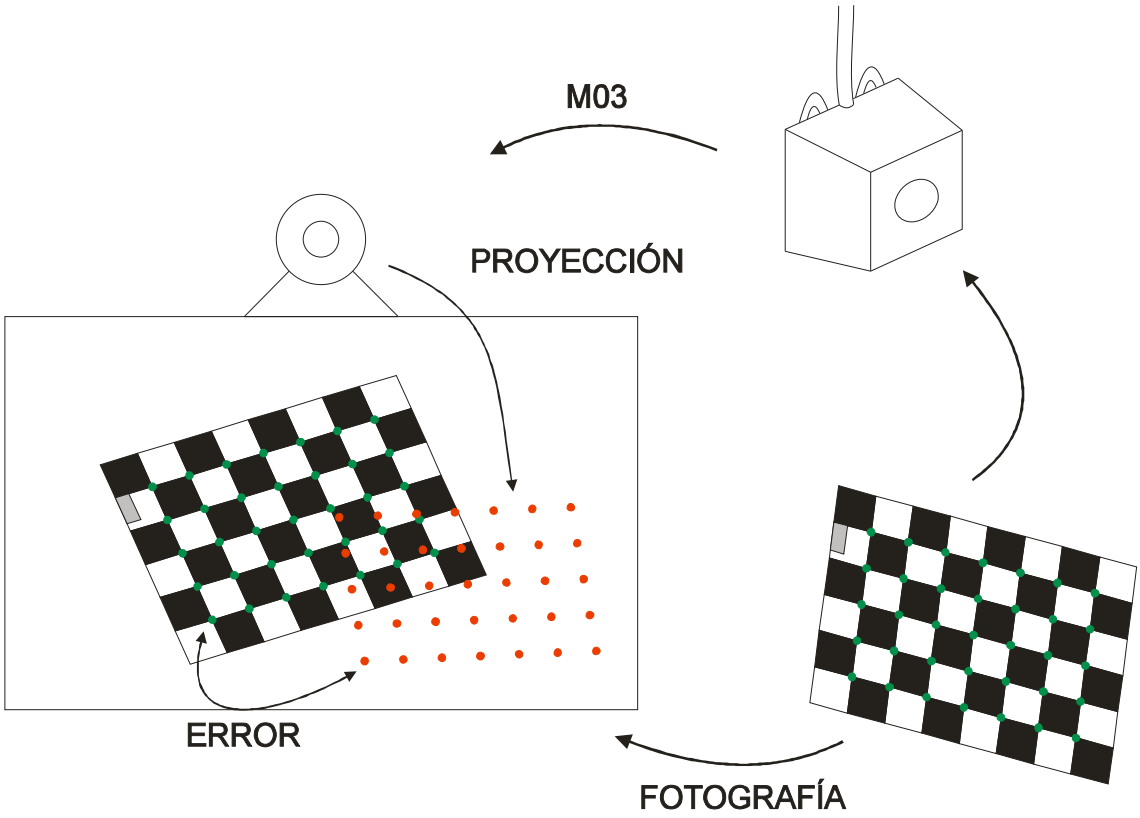


Fig. 6.7: Cálculo del error de proyección

Como conjunto de puntos espaciales para la optimización se han utilizado los vértices del damero. Una ventaja de utilizar estos puntos es que forman una cuadrícula perfecta, de manera que, conociendo las dimensiones de un cuadrado, el cálculo de las coordenadas de todos los vértices en el sistema de referencia del damero es inmediato. La otra ventaja está en que la localización de los vértices en la imagen es automática. De esta manera, quedan localizados los vértices en el sistema de coordenadas del damero y en la imagen sin ninguna dificultad, ya sólo queda proyectar y calcular el error.

Para proyectar los puntos es necesario conocer la relación damero – transmisor, para conocer las coordenadas de los vértices en el sistema de referencia del transmisor, y después utilizando la matriz M03 transformarlos a coordenadas de la cámara. Para ello se utiliza el sensor que está colocado en el damero. El sensor se relaciona con el transmisor mediante la matriz M10, y el damero se relaciona con el sensor mediante la matriz M21. Combinando esas dos matrices, más la matriz M03, ya es posible proyectar los vértices del damero en la imagen (Fig. 6.8).

Recordemos que la finalidad de todo este proceso es obtener la relación sensor – damero, es decir, las matrices M12 y M21, con la mayor precisión posible, es decir, que provoquen el menor error de proyección. Lo interesante de esto es que en el proceso de proyección se están utilizando las matrices M12 y M21, cuando todavía no se sabe cómo son, ya que son precisamente las que se quieren calcular. El método consiste en proyectar utilizando diferentes matrices M12 y M21 hasta encontrar aquellas que caracterizan de manera fiel la relación sensor-damero. Dicho de otra manera, una función de optimización proyecta los puntos del damero en la imagen, buscando aquella rotación y traslación sensor – damero, aquellas matrices M12 y M21, que minimizan el error de proyección.

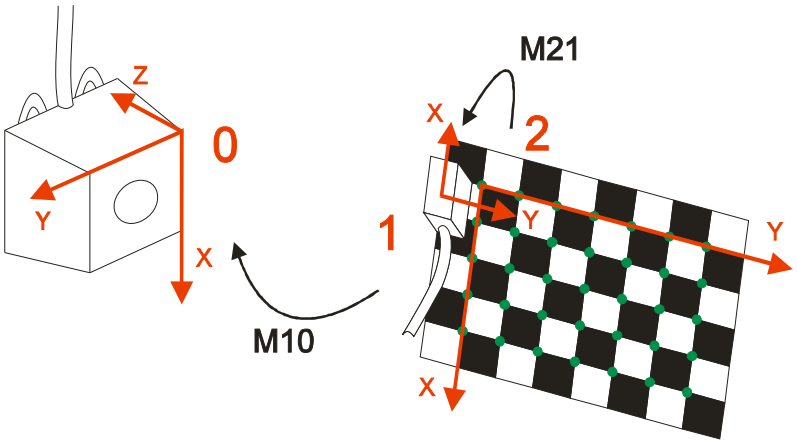


Fig. 6.8: Corrección del error de marcado

Este proceso de optimización supone un tiempo añadido de 5 minutos de procesado, un tiempo asumible si se observa los resultados obtenidos antes y después de la calibración. En la figura 6.9 se muestra 2 imágenes del sujeto de pruebas. El sujeto tiene una serie de marcas de referencia que se han marcado en las imágenes con cruces verdes. Esos puntos de referencia se han localizado tridimensionalmente con respecto al transmisor y se han proyectado en el plano imagen, las cruces rojas. La distancia entre las cruces verdes y rojas es el error de proyección.

Fig. 6.9: Error de proyección antes (arriba) y después (abajo) de la optimización

En las imágenes de la fila superior la proyección se ha realizado sin corregir el error de la matriz M12, y se observa un error evidente, de tal magnitud que no se pueden utilizar los puntos proyectados para ningún tipo de procesado. En las imágenes de la fila inferior la proyección se ha realizado con la matriz M12 corregida por optimización. La mejora en los resultados es notable.

El error de proyección calculado en un conjunto amplio de imágenes es el siguiente:

	Media (px)	Desviación estándar (px)
Sin corrección por optimización	6.2	2.8
Con corrección por optimización	1.5	1.0

Lo más interesante de llevar a cabo la optimización de esta manera es que se utilizan únicamente los datos tomados para la calibración, de modo que no es necesario ningún esfuerzo adicional por parte de la persona que calibra el sistema. Simplemente es necesario esperar unos minutos mientras el sistema se optimiza.

6.7 - Análisis de la cantidad de imágenes de calibración

Una pregunta interesante que surge a la hora de calibrar es cuál es el número idóneo de imágenes que se deben usar para obtener los mejores resultados. Para resolver a esta pregunta se han tomado 50 imágenes y se han calculado los parámetros intrínsecos de la cámara en función del número de imágenes de calibración, comenzando en 2 (no es posible calibrar con una imagen) y terminando en 50. El objetivo es encontrar el número de imágenes a partir del cual los parámetros se vuelven constantes y aumentar el número no afecta a los resultados.

La figura 6.10 muestra la evolución de los coeficientes de distorsión, una visión general (izquierda) y una visión detallada de la fase estacionaria (derecha).

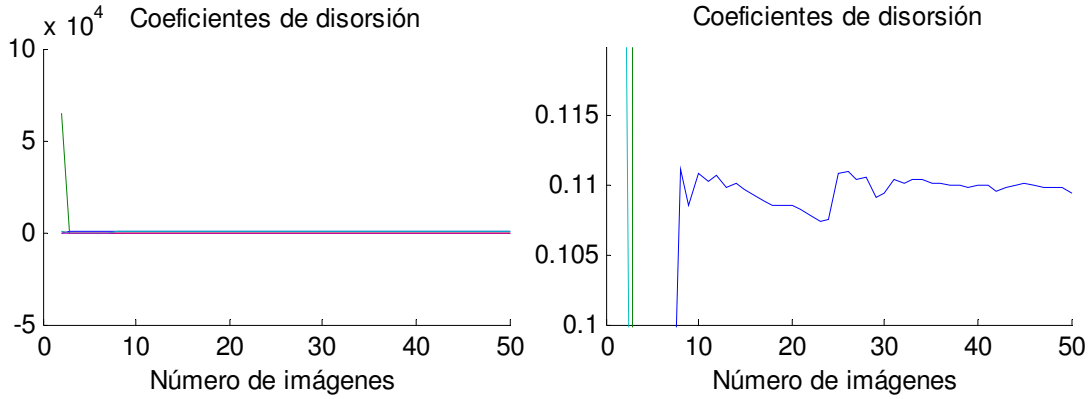


Fig. 6.10: Coeficientes de distorsión en función del número de imágenes de calibración

La figura 6.11 muestra la evolución de la distancia focal en los dos ejes, una visión general (izquierda) y una visión detallada de la fase estacionaria (derecha).

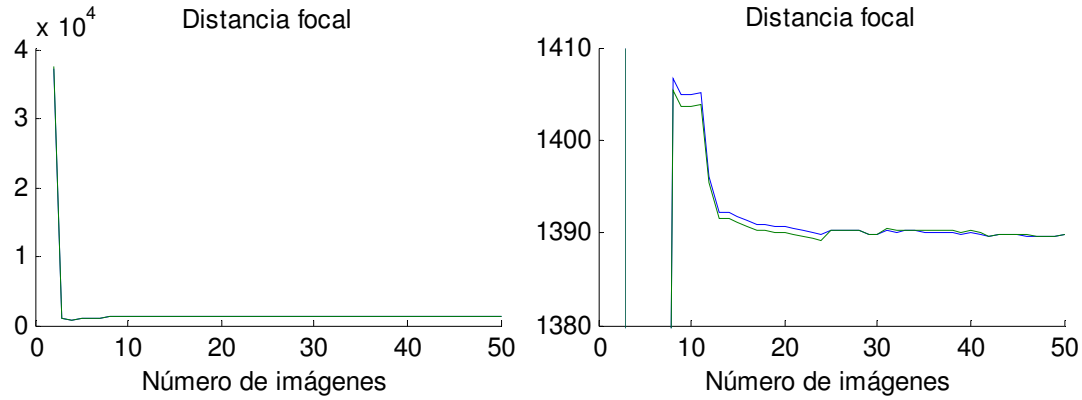


Fig. 6.11: Distancia focal en función del número de imágenes de calibración

La figura 6.12 muestra la evolución del punto principal, en el eje x (izquierda) y en el eje y (derecha).

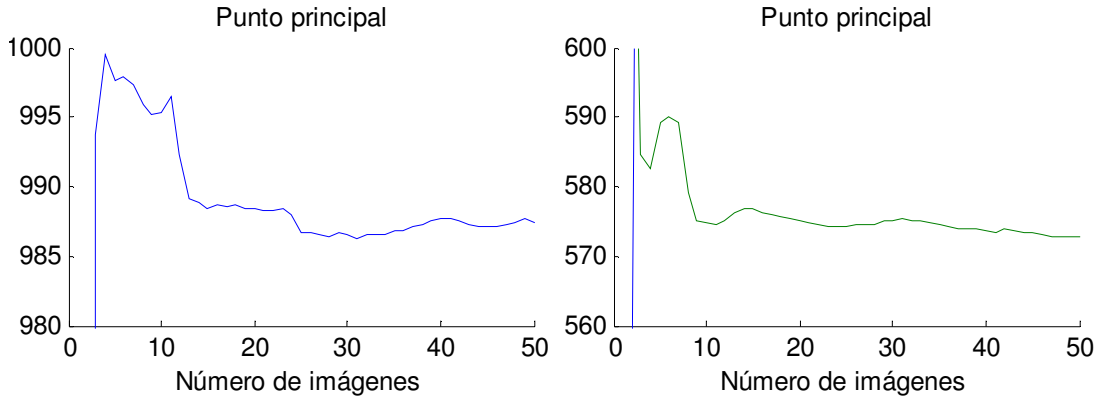


Fig. 6.12: Punto principal en función del número de imágenes de calibración

Se observa que al calibrar con pocas imágenes los parámetros calculados son erróneos, y sólo con un número de imágenes superior a 10 los valores comienzan a estabilizarse. Entre 10 y 30 imágenes los parámetros son bastante constantes pero se observan ciertas variaciones. Entre 30 y 50 imágenes los parámetros alcanzan un estado muy estable, bastante mejor que entre 10 y 30. En base a ello se opta por realizar la calibración con 50 imágenes, de manera que los parámetros obtenidos son muy fiables y el tiempo que requiere por parte del usuario es de aproximadamente 30 minutos.

El tiempo de calibración (tiempo que tarda el ordenador en calibrar la cámara a partir de las imágenes) aumenta de manera lineal con la cantidad de imágenes utilizadas. Una calibración con 50 imágenes supone un tiempo de procesado de 10-11 minutos, totalmente asumible teniendo en cuenta que es un proceso automático ajeno a la persona que calibra el sistema (Fig. 6.13).

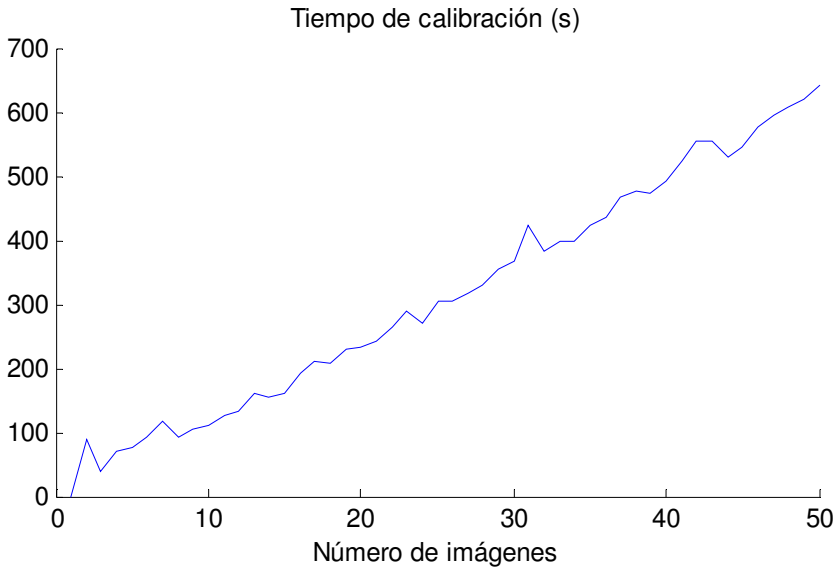


Fig. 6.13: Tiempo de calibración en función del número de imágenes de calibración

6.8 - Datos de calibración

Los parámetros intrínsecos de la cámara obtenidos son:

Resolución (px)	1920	1080
Distancia focal (mm)	1391.8	1390.8
Punto principal (px)	987.0	571.8
Distorsión radial	0.111	-0.185
Distorsión tangencial	-0.002	0.001
Skew (º)	0	
FOV (º)	78	

6.9 - Conclusiones

En resumen, en este proyecto se ha automatizado y mejorado la calibración del sistema transmisor-cámara.

Por un lado se ha simplificado todo lo referente a la toma de datos del damero y el sensor, de manera que con una mínima interacción del usuario se pueden tomar muchos datos en poco tiempo.

Por otro lado, se ha automatizado el proceso de calibración de manera que el usuario que realiza la calibración no debe hacer nada más que ejecutar el programa creado para tal tarea y esperar a que termine.

Además, se ha optimizado el cálculo de la transformación sensor-damero para minimizar el error de calibración.

Estos cambios suponen que el tiempo de una calibración con 50 imágenes disminuye desde 2-3 horas utilizando el método antiguo hasta 30 minutos utilizando los nuevos programas implementados en este trabajo (más 15 minutos de procesado automático que no requieren interacción de la persona que calibra el sistema). Además de la disminución en la duración de la calibración, el error de calibración se reduce a sólo un 24% del error original.

7 – Marcado de las imágenes

7.1 - Viabilidad del marcado manual de las imágenes

La base de datos se compone de 10 usuarios, con 12 videos por cada usuario y 300 *frames* por cada video, lo que supone un total de 36.000 *frames*. El tiempo medio de marcado manual de una imagen se estima en 5 minutos, de manera que el marcado de todas las imágenes supondría un total de 375 días en jornadas de 8 horas. Dado el alcance de este proyecto y el tiempo estimado de marcado, se concluye que el marcado manual de la base de datos es inviable.

En su lugar, se propone un método de marcado automático basado en el sistema sensor - transmisor, que por un lado elimina la tediosa tarea del marcado manual, y por otro supone un reto intelectual mucho más interesante.

7.2 - Sistema de marcado

Partiendo de la base de que durante la grabación el sensor de la cabeza de la persona permanece solidario a la cabeza, se trata de buscar espacialmente los puntos de la cara que interesa marcar y referenciarlos al sistema de coordenadas del sensor. Una vez localizados esos puntos, y sabiendo la localización y orientación del sensor, se proyectan esos puntos en la imagen y se obtiene el conjunto de puntos que identifican las regiones faciales de interés en coordenadas de imagen.

Dado un punto P, de coordenadas (x, y, z) en el sistema de referencia del sensor (1), se puede calcular la posición de P en el sistema de referencia del transmisor (0), utilizando la matriz de transformación M10. Del mismo modo, conociendo la matriz de cambio de sistema transmisor – cámara M03, se puede obtener la posición de P en el sistema de coordenadas de la cámara. Finalmente, sabiendo los parámetros intrínsecos de la cámara es posible proyectar P en el plano imagen, obteniendo el punto P’ de coordenadas de imagen (x’, y’) (Fig. 7.1).

La idea del marcado automático consiste en utilizar este método de proyección de un punto dado con respecto al sensor con los 54 puntos de interés. Una vez conocida la localización de los 54 puntos con respecto al sensor 1, es posible proyectarlos en todos y cada uno de los frames del video. Lo interesante de este método es que los 54 puntos van referenciados al sensor 1, de modo que cuando el sensor se mueve los puntos se mueven con él, y su proyección coincide con las de los puntos faciales característicos. La dificultad radica en obtener la localización espacial de los 54 puntos en el sistema de referencia del sensor.

Por otro lado, si se dispone de 2 sensores y se conoce la posición espacial de un punto P con respecto a uno de ellos (sensor 2), mediante matrices de cambio de sistema de referencia se puede obtener las coordenadas del punto P en el sistema de referencia del otro sensor (sensor 1). Entonces, situando el sensor 1 en la cabeza del sujeto, y localizando los puntos de interés con el sensor 2, se puede obtener la posición de esos puntos en el sistema de referencia del sensor 1 (Fig. 7.2), y posteriormente proyectarlos sobre la imagen (Fig. 7.1).

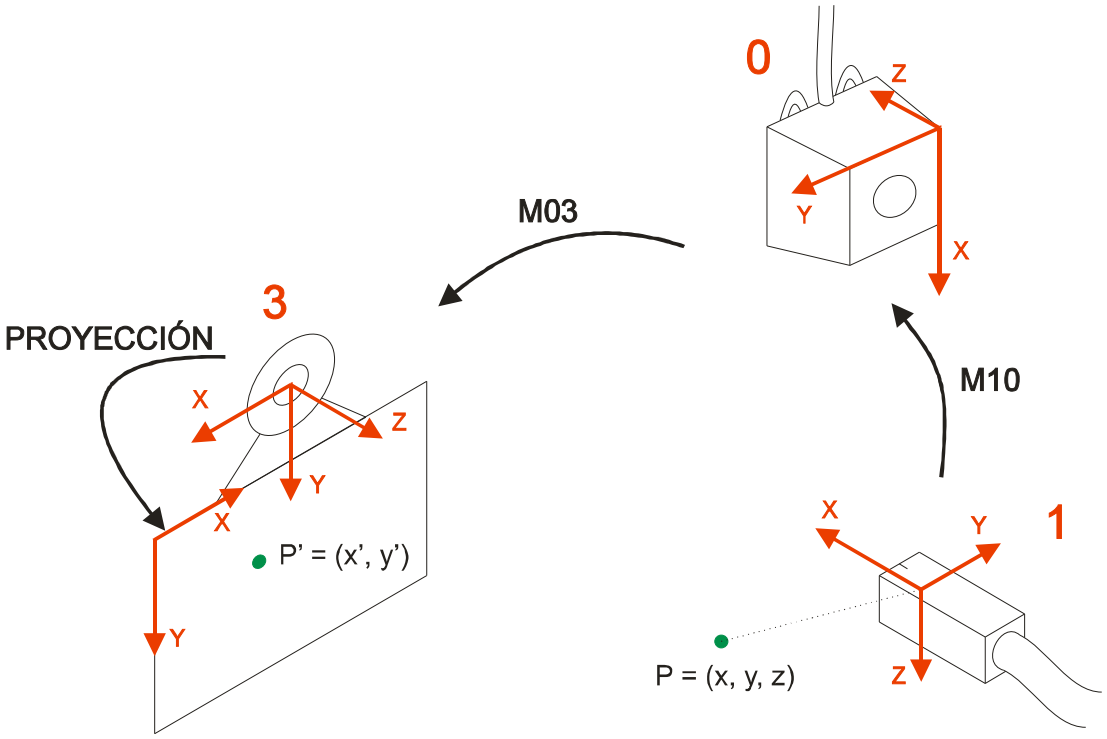


Fig. 7.1: Proyección de un punto P en el plano imagen

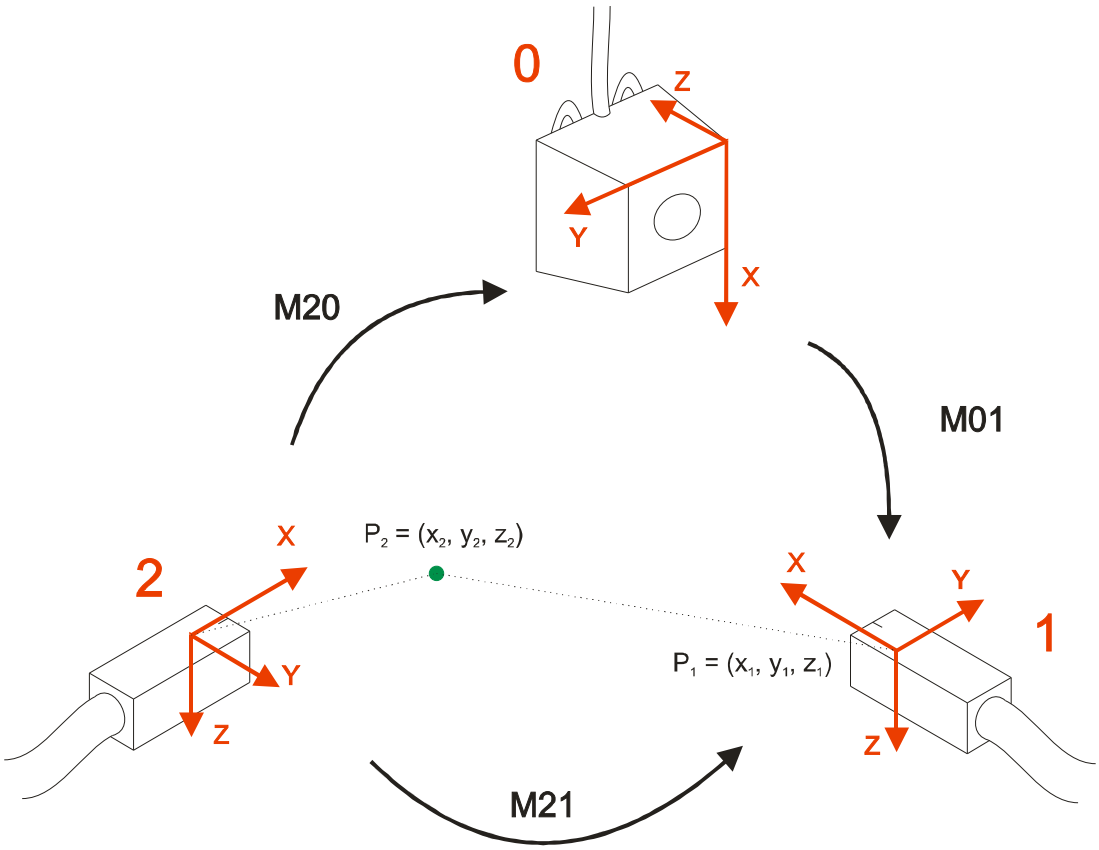


Fig. 7.2: Cambio de sistema de coordenadas de un punto entre dos sensores

Para la localización de los puntos con respecto al sensor 2 se ha utilizado un útil denominado marcador, una pieza rígida con un extremo acabado en punta. Dado que se conoce las dimensiones del útil, la punta se encuentra en un punto conocido y constante con respecto al sensor 2. Si esa punta se sitúa sobre el punto facial que interesa localizar, ese punto facial pasa a estar en un punto conocido con respecto al sensor 2, P_2. Con las transformaciones anteriores se calcula la localización del punto facial con respecto al sensor 1, P_1.

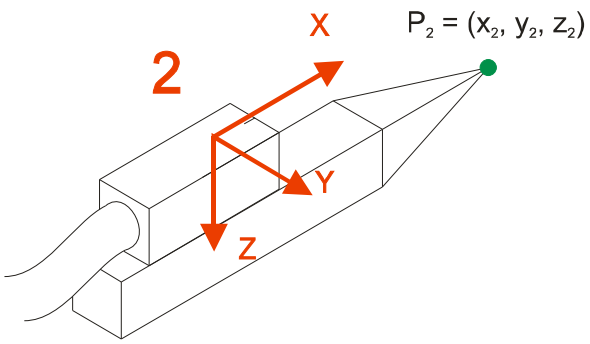


Fig. 7.3: Localización de un punto frente respecto al sensor 2

La localización de la punta del marcador es a priori sencilla, dado que se conoce el lugar exacto del origen del sensor alojado en ella y se conoce sus dimensiones. La realidad es que debido a los márgenes de error de la impresora 3D con la que fue creada existe cierta incertidumbre en su localización. Para estimar la punta del marcador se ha llevado a cabo un proceso de optimización.

Dado un punto espacial fijo con respecto al transmisor, el punto verde de la figura 7.4, se sitúa la punta del marcador en ese punto y se mueve el marcador en todas direcciones, sin apartar la punta de ese punto, mientras se captura datos sensor-transmisor. El origen del sensor, y en general todos los puntos del marcador, están en constante movimiento con respecto al transmisor, pero hay un único punto P_2, en el sistema de referencia del sensor, que permanece estático con respecto al transmisor. Utilizando una función de optimización se busca las coordenadas de ese punto (x_2, y_2, z_2) con respecto al sensor en movimiento de manera que su variabilidad a lo largo de ese movimiento con respecto al transmisor sea mínima. O lo que es lo mismo, el único punto perteneciente a la pieza cuyo movimiento sea mínimo.

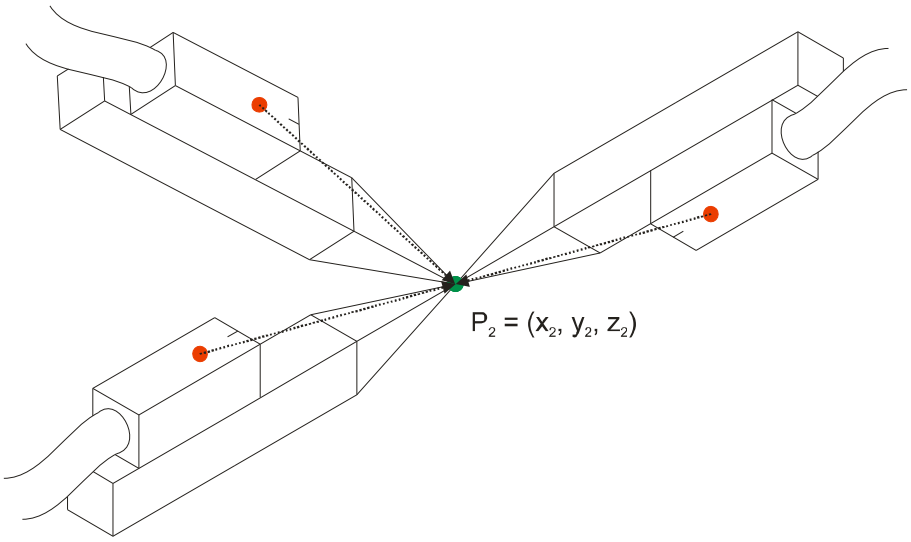


Fig. 7.4: Localización de la punta del marcador

Repitiendo el marcado en los 54 puntos, se obtiene un conjunto de puntos tridimensionales, que representan la posición de los 54 puntos faciales, referenciados al sensor 1, en la cabeza del sujeto (Fig. 7.5).

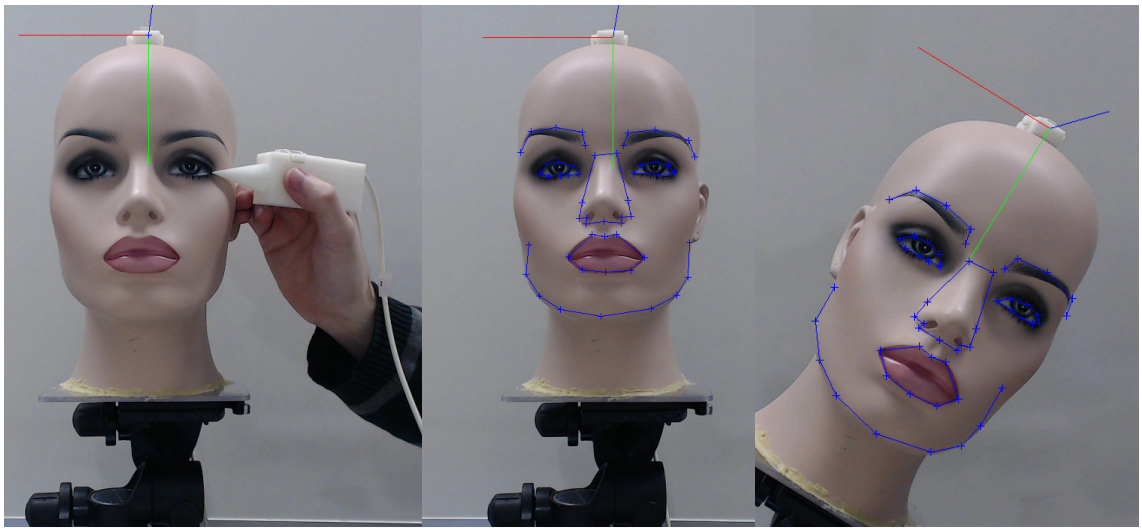


Fig. 7.5: Proceso de marcado (izquierda) y localización de los 54 puntos del marcado (centro y derecha)

Debe remarcarse que los 54 puntos están referenciados al sensor, y en ningún momento se realizan marcas físicas en el sujeto de ningún tipo, sólo artificios matemáticos (Fig. 7.6). Además, durante el marcado la persona tiene libertad de movimientos, ya que cualquier movimiento de su cabeza se traduce en el movimiento correspondiente del sensor, y en el sistema de referencia del sensor todo permanece en el mismo lugar.

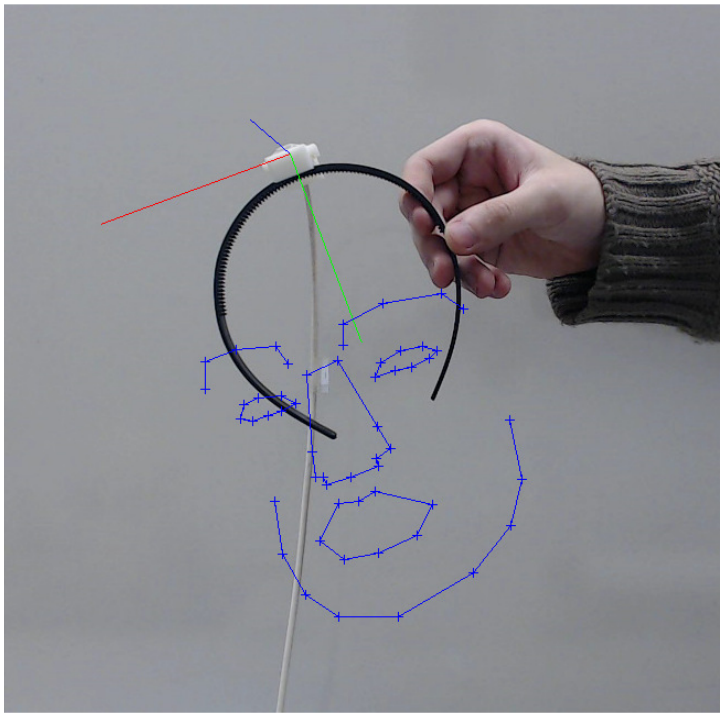


Fig. 7.6: Las marcas están referenciadas al sensor

7.3 - Piezas utilizadas

Las piezas utilizadas durante el proceso de marcado y grabación se han diseñado y construido expresamente para este proyecto.

La primera pieza, denominada marcador, se utiliza para la localización de los puntos faciales del sujeto. Aloja el sensor 2, y tiene un extremo acabado en punta para la localización precisa del punto que se desea marcar (Fig. 7.7 a).

La segunda pieza se utiliza para colocar el sensor 1 en la cabeza artificial con apariencia real. La base de la pieza tiene la forma de la cabeza de manera que su ajuste es perfecto (Fig. 7.7 b). Esto permite realizar múltiples pruebas durante el desarrollo del sistema sin depender de sujetos reales.

La tercera pieza se utiliza para colocar el sensor 1 en la cabeza del sujeto de grabación. Tiene una ranura en su base para unirla a una diadema y sujetarse a la cabeza de manera sólida (Fig. 7.7 c).

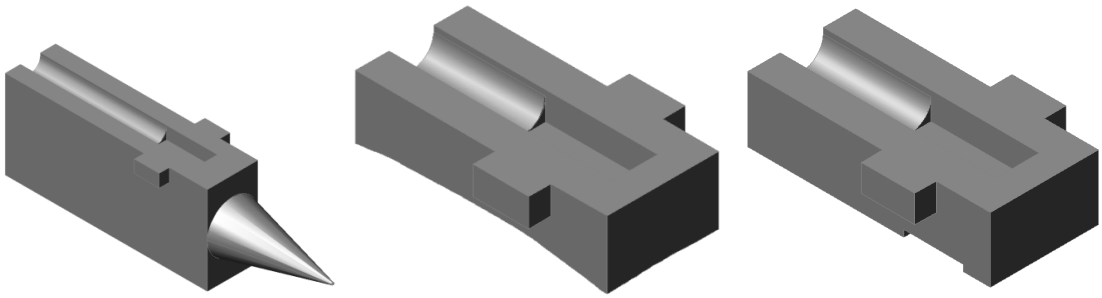


Fig. 7.7: a) marcador b) pieza para cabeza artificial c) pieza para cabeza real

7.4 - Error de marcado

Para calcular el error del marcado se ha comparado el modelo obtenido con el marcador y el conjunto sensor-transmisor con el modelo obtenido con un escáner tridimensional profesional.

Por un lado se ha marcado los 54 puntos en el modelo de cabeza (Fig. 7.5), obteniendo como resultado una nube de puntos tridimensional con la forma de la cabeza. Por otro lado se dispone de un escaneado tridimensional de la misma cabeza, realizado con un sistema profesional de gran precisión [8].

Mediante un proceso de registro se ajustan las marcas manuales al modelo tridimensional, y se mide la distancia entre los puntos y el modelo, en unidades espaciales, milímetros en este caso.

El error del marcado promedio es de 0.7 mm, con una desviación estándar de 0.3 mm.

8 – Grabación de los vídeos

La grabación se ha realizado en líneas generales de manera análoga a la descrita por [1]. Sin embargo, al igual que se ha hecho con la calibración, se han añadido algunas funcionalidades al sistema de grabación y se ha simplificado su manejo.

8.1 - Interfaz gráfica de grabación

Para dotar al sistema de grabación de una mayor simplicidad se ha remplazado la función de grabación a base de comandos por una interfaz gráfica. Los botones están dispuestos siguiendo el mismo orden en el que se pulsan en una grabación normal, de este modo con unas nociones básicas de manejo cualquier persona puede utilizar el sistema de grabación sin necesidad de conocer todos los procesos que ocurren por detrás (Fig. 8.1).

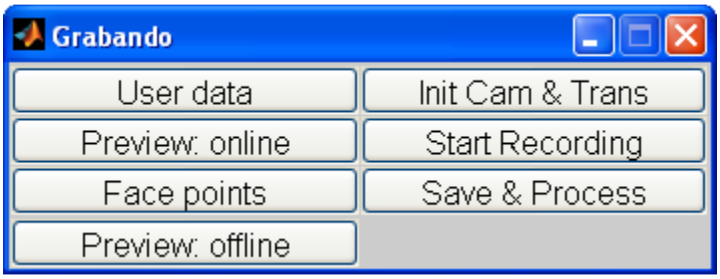


Fig. 8.1: interfaz gráfica para la grabación

Un proceso típico de grabación sigue el siguiente orden:

User data:
Antes de comenzar con la grabación se introduce información sobre el sujeto de grabación, como el nombre, fecha de nacimiento o centro de estudios. En este proceso se asigna además un número al usuario. Una vez introducida la información del sujeto se genera automáticamente una carpeta que contendrá los datos de la grabación, como los videos, datos del sensor o las 54 marcas faciales, así como los datos de la calibración utilizada en esa grabación (parámetros intrínsecos de la cámara y matrices M03 y M30).

Preview online:
El siguiente paso consiste en colocar la diadema con el sensor al sujeto. Esta colocación no es trivial, sino que debe ser aquella que sitúe al sensor en una orientación completamente frontal con respecto a la cámara, con una rotación de 0º en los 3 ejes, cuando la cabeza de la persona está totalmente frontal respecto a la cámara. Para ello se ha implementado una función que muestra en pantalla la información de la cámara y el sensor en tiempo real. En el caso de haber marcado la cara del usuario previamente, los puntos se ven proyectados en la imagen (Fig. 8.2).

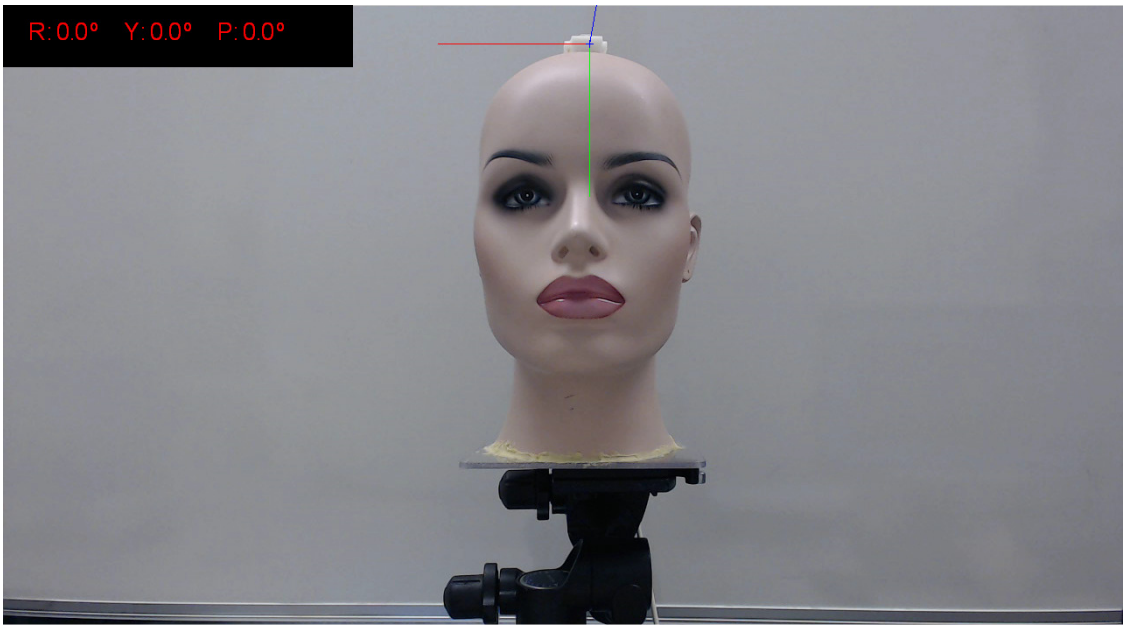


Fig. 8.2: imagen de *preview*, se muestra la orientación de la cabeza

Face points:
Una vez situada la diadema en la cabeza de manera correcta se procede a marcar los 54 puntos faciales por el método descrito anteriormente. El sistema permite repetir el marcado de uno o varios puntos tantas veces como se desee. El resultado de ese marcado, una nube tridimensional de puntos referenciados al sistema de coordenadas del sensor de la cabeza (sensor 1), se almacena en el archivo user_XX_face_s1.mat. Si tras el marcado de los puntos se vuelve a utilizar la función de *preview online*, además de la orientación de la cabeza se verá en la imagen la proyección de los puntos marcados (Fig. 8.3). De esta manera es posible analizar la calidad del marcado y corregir algunos puntos en caso de que se crea conveniente.

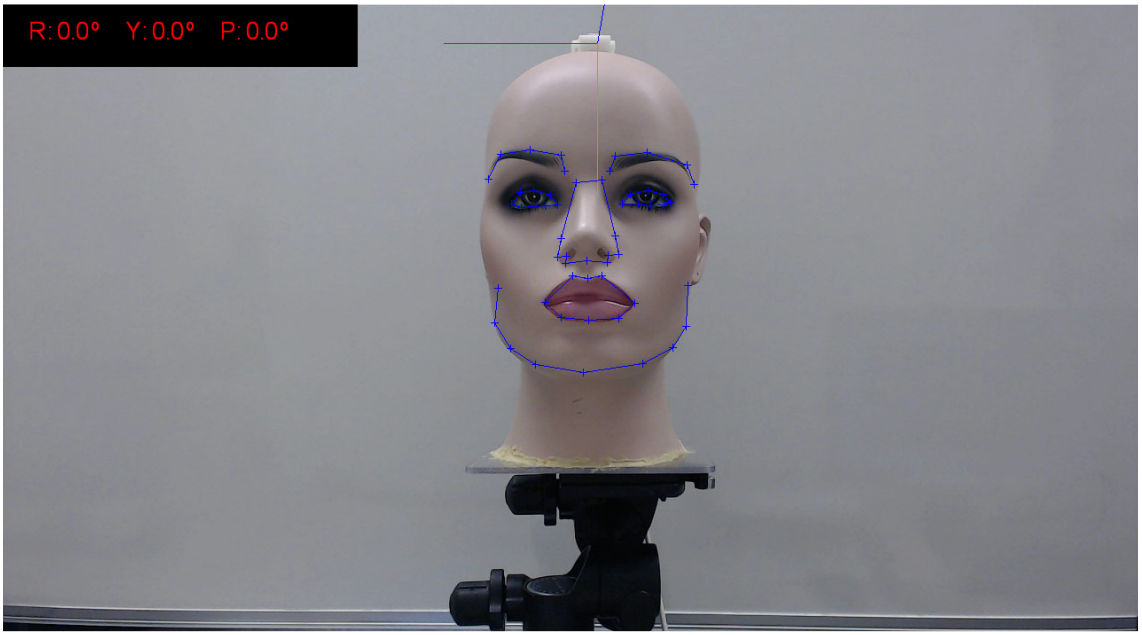


Fig. 8.3: imagen de *preview*, se muestra la orientación de la cabeza y los puntos marcados

Preview offline:
La función *preview offline* es una variante del *preview online* que muestra solamente una imagen, con la orientación de la cabeza correspondiente, y la proyección de los puntos en el caso de que se haya marcado la cara previamente. El hecho de usar una única imagen permite analizar con más detalle la precisión del marcado, sin la molestia de los movimientos del sujeto durante el análisis.

Init Cam & Trans:
El primer paso de la grabación consiste en inicializar la cámara. En este caso se ha configurado una grabación a resolución de 1280x720 px, 30 *frames* por segundo y 300 *frames* en total, con enfoque manual. El siguiente paso consiste en iniciar el sistema de adquisición trakSTAR. Tras un proceso de inicialización que oscila entre 5 y 10 segundos, comienza la adquisición de los datos.

Start Recording:
Tras comprobar que el sistema trakSTAR está adquiriendo datos, se inicia la grabación del vídeo. Hay que ser especialmente cuidadoso en esta fase, ya que en el caso de comenzar la grabación de vídeo sin esperar a que el sistema trakSTAR comience la adquisición, algunos frames no tendrán información sobre la orientación de la cabeza y se deberá descartar el video.

Save & Process:
Una vez finalizada la grabación del video y la adquisición de datos del sensor, se guardan los datos de la grabación en la carpeta del usuario, típicamente el vídeo y los datos del sensor con respecto al transmisor. Inmediatamente después se procesa el video. El procesado consiste en estimar, a partir de los datos del sensor con respecto al transmisor y las matrices de calibración, la posición y orientación del sensor (cabeza del usuario) con respecto a la cámara. Además, se proyectan los puntos del marcado sobre el plano imagen. Estos datos de orientación-posición 3D de la cabeza y los *landmarks* obtenidos mediante proyección componen el *ground truth*.

8.2 - Procesado

El procesado del video consta de dos partes. La primera consiste en calcular la posición y orientación del sensor en el sistema de coordenadas de la cámara, y la segunda consiste en proyectar los puntos tridimensionales de la cara del usuario en el plano imagen (Fig. 8.4).

Dada una adquisición del sensor con respecto al transmisor, se calcula la matriz de cambio de sistema de referencia M10. Conociendo la matriz de cambio de coordenadas M03 por la calibración, se calcula M13 como M03 * M10, que representa la matriz de transformación del sensor a la cámara. De esa matriz se obtiene la rotación y traslación del sensor con respecto a la cámara.

El objetivo deseado es que cuando el sensor se sitúa frente a la cámara en una posición totalmente frontal el vector de rotación sea (0,0,0). Sin embargo, esto no ocurre así, porque los ejes de coordenadas de la cámara son diferentes a los ejes del sensor. El resultado de esta diferencia es que con el sensor totalmente frontal se obtiene un vector de rotación (-90,0,90). Es preciso pues corregir el sistema de ejes del sensor para adecuarlos al sistema de ejes de la cámara. Para ello se define un sensor imaginario (4) en la misma posición que el sensor 1 pero con los ejes cambiados y se construye la matriz de cambio de ejes M41.

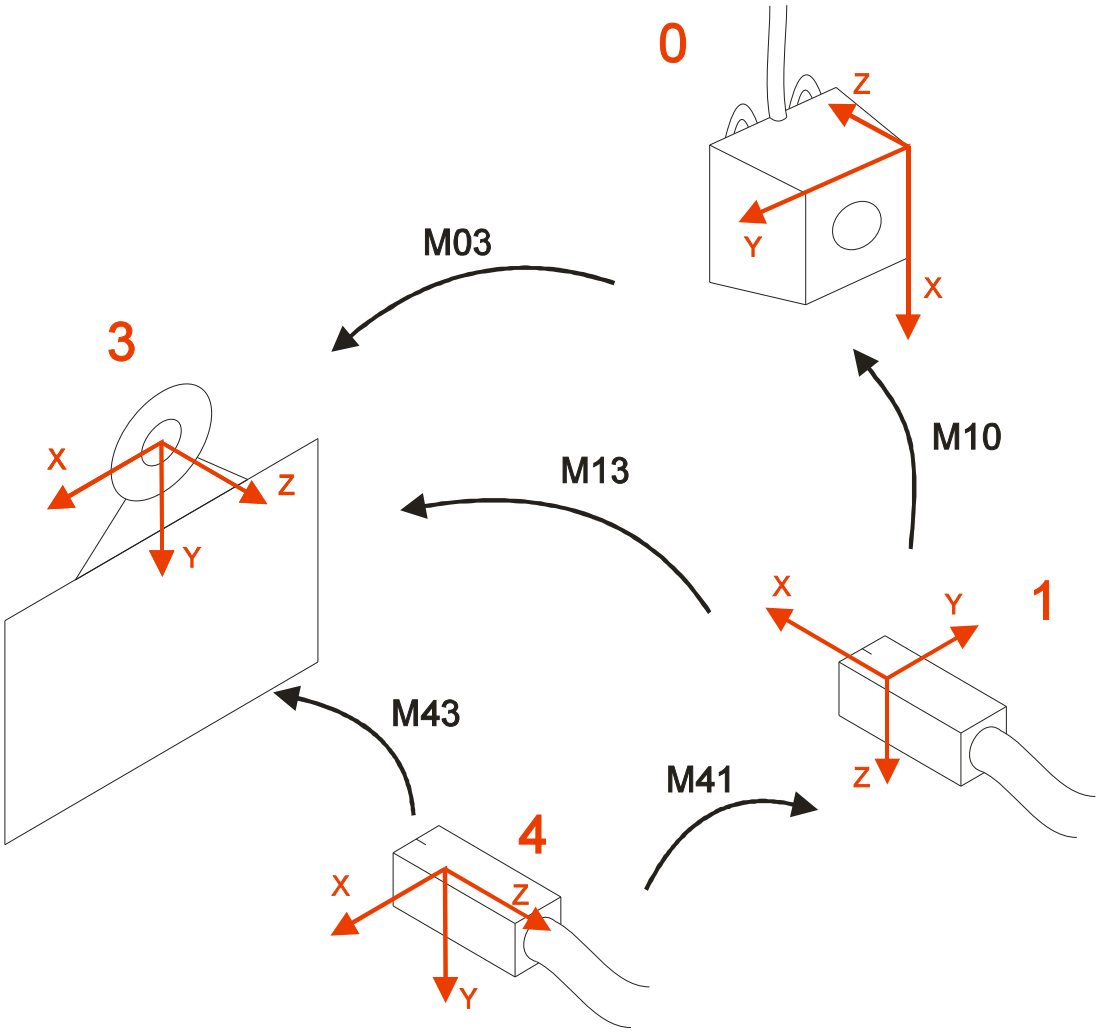


Fig. 8.4: elementos del procesado del video

$$M41 = \begin{pmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

De esta manera se calcula la matriz de transformación sensor 4 – cámara como:

M43 = M13 * M41 = M03 * M10 * M41

De la matriz M43 se obtiene la posición y orientación del sensor con respecto a la cámara, pero con un sistema de ejes tal que cuando el sensor está totalmente frontal el vector de rotación sí es (0,0,0).

Para la proyección de los puntos faciales marcados se calcula la matriz M13 de transformación sensor 1 – cámara y se proyectan utilizando esa matriz y los parámetros intrínsecos de la cámara. Dado que los puntos marcados tridimensionalmente están referenciados al sensor 1, no se utiliza el cambio de ejes de dicho sensor.

8.3 - Corrección de la proyección de los puntos

Un problema que se ha dado durante la grabación de algunos videos es que desde el marcado inicial hasta la grabación del último video la diadema con el sensor se mueve ligeramente y los puntos proyectados se desplazan también ligeramente de su localización real en la imagen.

Para solucionarlo se ha optado por realizar un nuevo marcado al finalizar la grabación, y combinar los dos marcados para obtener un modelo que se adapte mejor a la cara del usuario, compensando de este modo el movimiento del sensor. El nuevo modelo se calcula como una combinación lineal de ambos:

$$Cara\_nueva = \alpha * Cara\_inicial + (1-\alpha) * Cara\_final, \qquad 0 \leq \alpha \leq 1$$

Para esta tarea se ha implementado una función con una interfaz gráfica que permite combinar los dos marcados y visualizar el resultado sobre las imágenes del video al instante. La combinación de marcado se realiza a nivel de órganos, es decir, un ojo es independiente del otro, e independiente de las cejas o la nariz (Fig. 8.5). Una vez elegida la combinación de modelos para todos los órganos se procesa el video y se obtiene el archivo con las marcas 2D.

Fig. 8.5: interfaz para la corrección del movimiento del sensor

8.4 - Montaje e iluminación

Para la grabación de los vídeos se ha empleado iluminación artificial, de manera que es posible controlar en todo momento la iluminación de la persona que se está grabando, evitando las constantes variaciones de intensidad de la luz solar. De esta manera se ha conseguido una iluminación semejante para todos los sujetos de la base de datos.

La iluminación utilizada es luz difusa, cuyo objetivo es iluminar la cara de manera homogénea evitando la aparición de sombras. Para ello se han utilizado 2 focos de estudio junto con un conjunto reflector-difusor para convertir la luz directa de los focos en luz difusa. Además, un tercer foco ilumina el fondo de la escena. El montaje se muestra en la figura 8.6.

La cámara se sitúa frente al difusor, a aproximadamente 40 cm de distancia. A esa distancia la cámara no provoca ningún tipo de sombra en la cara de la persona. La persona se sitúa frente a la cámara a una distancia de 60 cm. El transmisor se ha colocado sobre el sujeto, a aproximadamente 30 cm sobre su cabeza, distancia lo suficientemente lejos como para que la persona no lo golpee durante la grabación pero lo suficientemente cerca como para que la calidad de la medida no se vea afectada por la distancia.

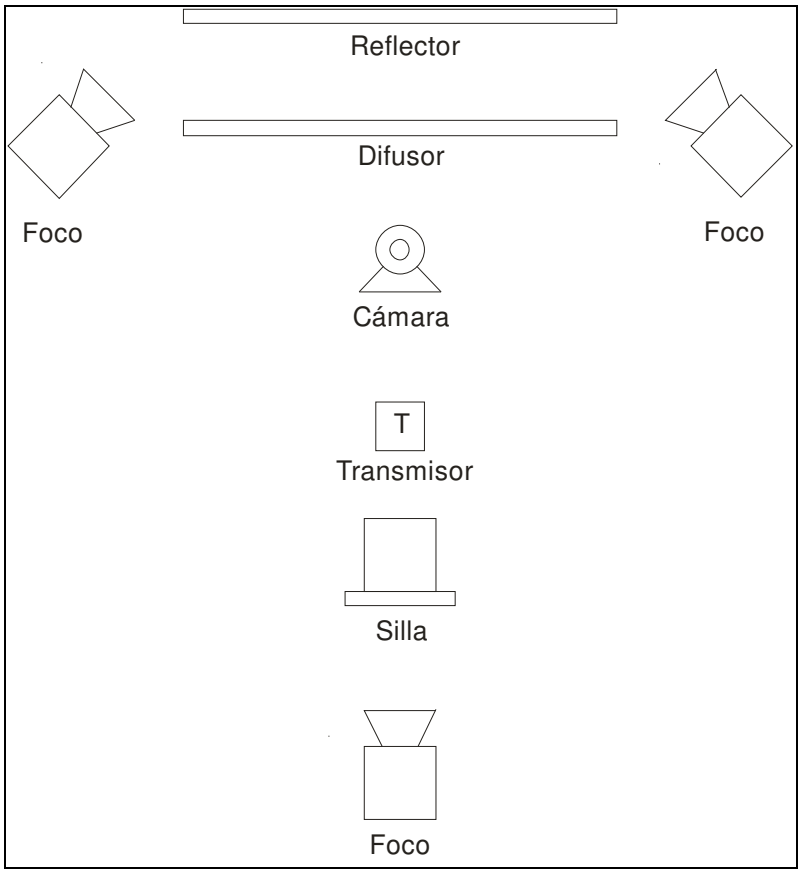
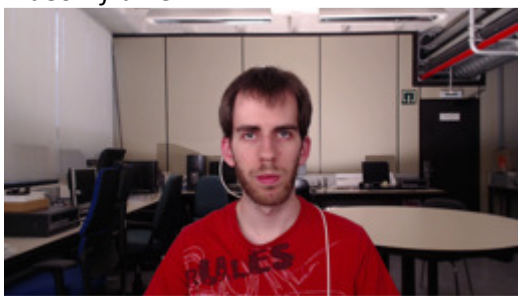


Fig. 8.5: sistema de iluminación y grabación

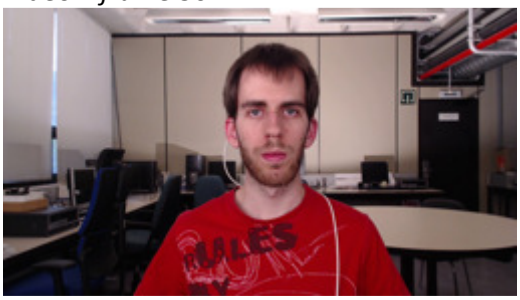
8.5 - Ejemplos

Se muestra a continuación un *frame* de cada video grabado al usuario 2, en diferentes posiciones.

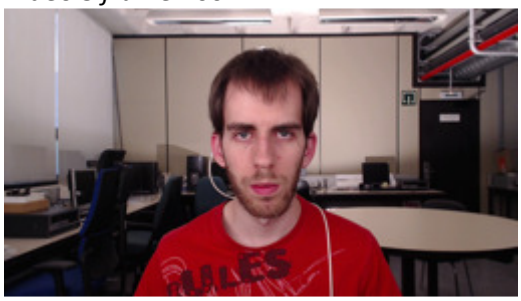
Video 1 *frame* 1



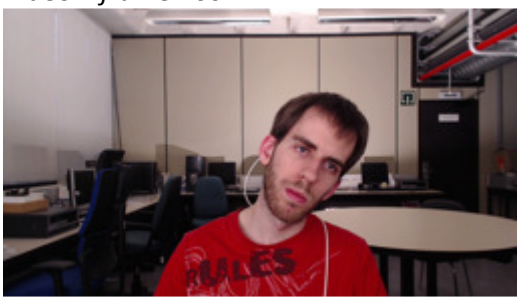
Video 2 *frame* 50



Video 3 *frame* 100

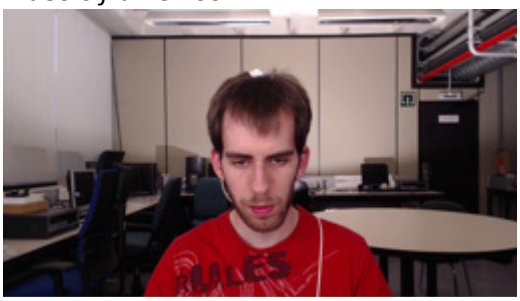


Video 4 *frame* 100

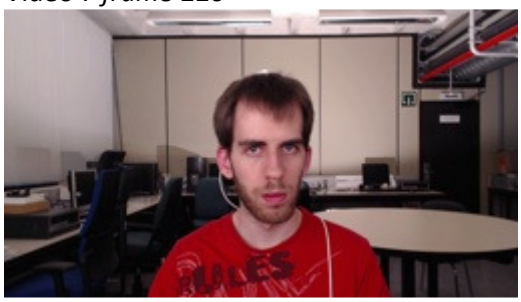


Video 5 *frame* 100

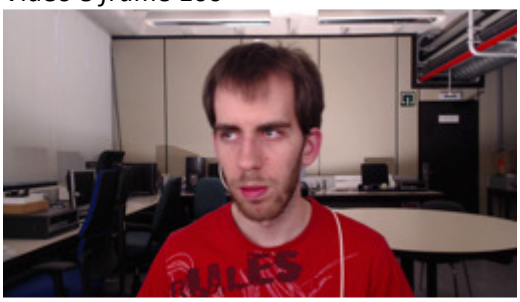
Video 6 *frame* 100



Video 7 *frame* 220

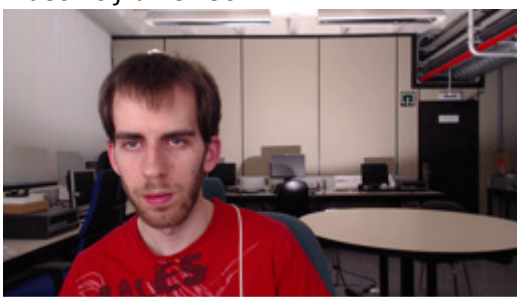


Video 8 *frame* 100

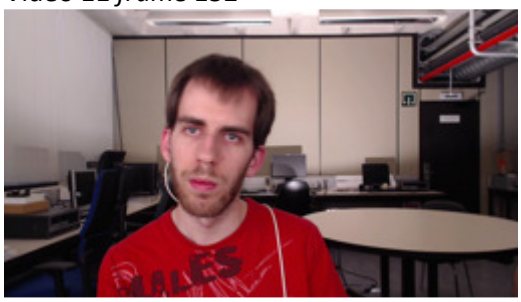


Video 9 *frame* 150

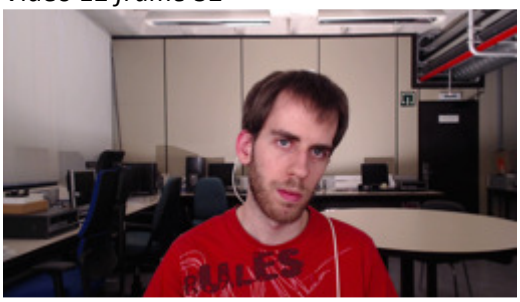
Video 10 *frame* 150



Video 11 *frame* 152

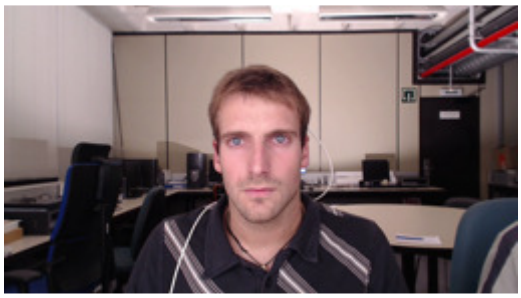


Video 12 *frame* 52

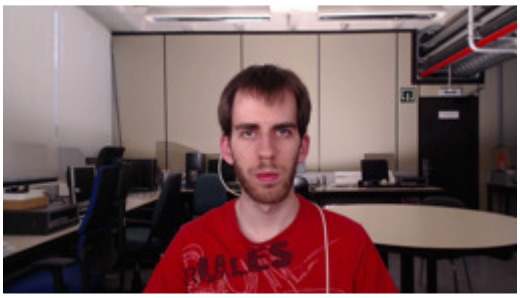


Se muestra a continuación el *frame* inicial del primer video de cada usuario.

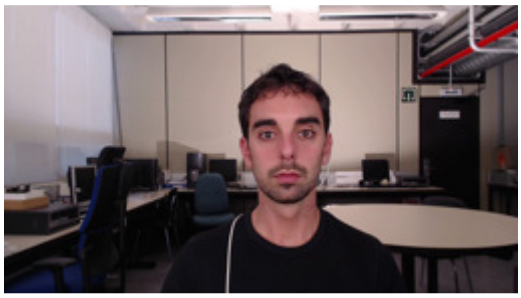
Usuario 1



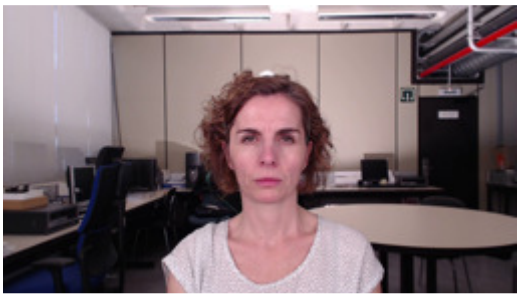
Usuario 2



Usuario 3

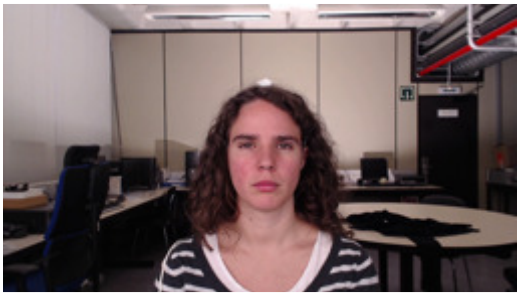


Usuario 4



Usuario 5

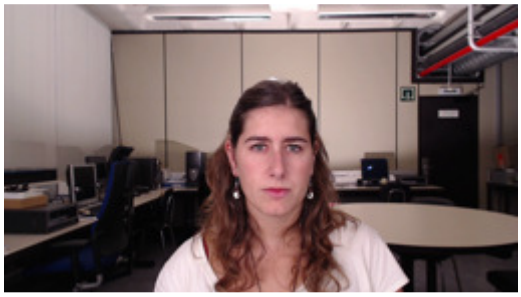
Usuario 6



Usuario 7

Usuario 8

Usuario 9



Usuario 10

9 – Estimación de la posición de la cabeza

9.1 - Introducción

La estimación de la posición de la cabeza (*head pose estimation* o HPE) [9] consiste en obtener, a partir de una imagen de la cara de una persona, la posición y orientación de la cabeza con respecto a la cámara. En este trabajo se comparan 4 métodos: una combinación de ASM y POSIT [10][11], algoritmos de segmentación y estimación de posición respectivamente; AAM y POSIT [12][13][14], otro algoritmo de segmentación junto con el mismo algoritmo de estimación de posición; FaceAPI [15], un sistema comercial desarrollado por SeeingMachines; e Intraface [16], un novedoso sistema desarrollado por HumanSensing. Los algoritmos se han analizado con la base de datos implementada en este trabajo, y se han comparado teniendo en cuenta la precisión en los resultados, la estabilidad ante situaciones estacionarias, y finalmente en cuanto a tiempo de procesado.

9.2 - ASM + POSIT

El primer método estudiado estima la posición de la cabeza en dos etapas. La primera etapa consiste en segmentar la imagen del usuario para identificar los 54 *landmarks*, utilizando un algoritmo denominado ASM [10]. La segunda etapa consiste en estimar, a partir de los 54 *landmarks* y un modelo tridimensional de la cabeza, la posición y orientación del usuario en esa imagen, para lo que se emplea el algoritmo POSIT [11].

ASM

ASM, *Active Shape Models* [10], es un método de segmentación de imágenes que combina la información de forma y apariencia de los objetos de un conjunto de imágenes de entrenamiento, desde un punto de vista estadístico, para encontrar en una nueva imagen los objetos que más se asemejan a los proporcionados en el entrenamiento. En este caso, los objetos mencionados son los órganos faciales. Es decir, es necesario construir un modelo que describa los objetos que se desea buscar para después buscar en una nueva imagen los objetos que mejor se ajusten a ese modelo.

En primer lugar se construye un modelo estadístico de forma. El objetivo es extraer de las imágenes de entrenamiento la mayor cantidad de información sobre la forma de los objetos presentes en ellas. Se toman los *landmarks* de esas imágenes y se aplica análisis de componentes principales (PCA) para reducir su dimensionalidad. De esta manera, se puede aproximar cada una de las caras de entrenamiento como:

$$x \approx \bar{x} + P \cdot b$$

donde $\bar{x}$ es la forma promedio y P es la matriz de k vectores propios de la matriz de covarianza, y b se define como:

$$b = P' (x - \bar{x})$$

La finalidad de aplicar PCA es caracterizar la forma de la cara usando b, un vector de dimensión k, menor que la dimensión del vector original, 108.

El modelo de apariencia se construye con la apariencia de la imagen alrededor de los *landmarks*. Para cada uno de los *landmarks*, se toma un perfil de apariencia de manera perpendicular al contorno del objeto y se normaliza para obtener un perfil de apariencia más robusto ante cambios de iluminación (Fig. 9.1). Una vez obtenidos los perfiles de apariencia de un *landmark* en todas las imágenes, se aplica PCA con el objetivo de caracterizar esa apariencia con la menor cantidad de datos.

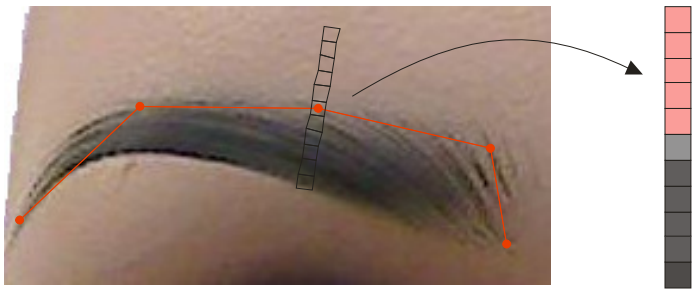


Fig. 9.1: perfil de apariencia perpendicular al contorno del objeto

Dada una imagen nueva, la segmentación consiste en localizar aquellos puntos de la imagen cuya forma es coherente con modelo estadístico de forma, y sus perfiles de apariencia son coherentes con el modelo estadístico de apariencia. Para cuantificar esa coherencia se utilizan los valores del vector b con la forma y la distancia de Mahalanobis con la apariencia.

En la implementación de ASM utilizada se realiza una segmentación multiresolución. Consiste en segmentar la imagen comenzando con un tamaño de imagen reducido, para hacer una primera detección grosera, e ir incrementado el tamaño de la imagen obteniendo segmentaciones cada vez más precisas hasta llegar al tamaño original de la imagen, resultando en una segmentación muy fina.

En cada nueva resolución se reduce el tamaño de la imagen a la mitad. Esto significa que en la resolución 1 se segmenta la imagen a su tamaño original. En la resolución 2 se reduce la imagen una vez y se segmenta, y se realiza otra segmentación en la resolución 1 partiendo del resultado de la segmentación anterior. En general, en la resolución i-ésima se comienza segmentando la imagen reducida i-1 veces y el resultado se utiliza para segmentarla nuevamente a la resolución i-1, y así sucesivamente hasta el tamaño original, realizando un total de i segmentaciones.

Dado que no hay un método para conocer en qué resolución comenzar la segmentación, se han segmentado todas las imágenes partiendo de 5 resoluciones diferentes (la original, 2, 3, 4 y 5), y comparado los resultados obtenidos con cada una de ellas.

Una vez segmentada la imagen se procesan los landmarks con el algoritmo POSIT.

POSIT

La posición de la cabeza se calcula utilizando el algoritmo POSIT (*Pose from Orthography and Scaling with Iterations*) [11], un método muy utilizado para la estimación de la posición de objetos 3D. La idea de POSIT consiste en que si se dispone de la proyección de un objeto en el plano imagen, y se dispone de las coordenadas del objeto en 3D, es posible estimar la posición y orientación del objeto en el espacio que, al proyectarse sobre el plano imagen, proporciona esa proyección 2D. El algoritmo calcula un vector de traslación T y una matriz de rotación R

que determinan la transformación entre el sistema de referencia del objeto, en este caso la cabeza, y el de la cámara. El vector de traslación representa la distancia entre la cámara y la cabeza. A partir de la matriz R se obtiene las rotaciones en los 3 ejes (Fig. 9.2).

El algoritmo asume un modelo de perspectiva débil para el cálculo de la posición y orientación. Se trata de una aproximación que considera que todos los puntos de modelo están en el mismo plano con respecto a la cámara. Esta aproximación permite una estimación en un tiempo de procesado mínimo (del orden del microsegundo) y el error que implica es despreciable.

Es necesario que exista correspondencia entre los puntos proyectados 2D y los puntos del modelo 3D, de modo que el punto i-ésimo del conjunto 2D se corresponda con el punto i-ésimo del conjunto 3D. De ahí la importancia de obtener localizaciones de landmarks conocidos.

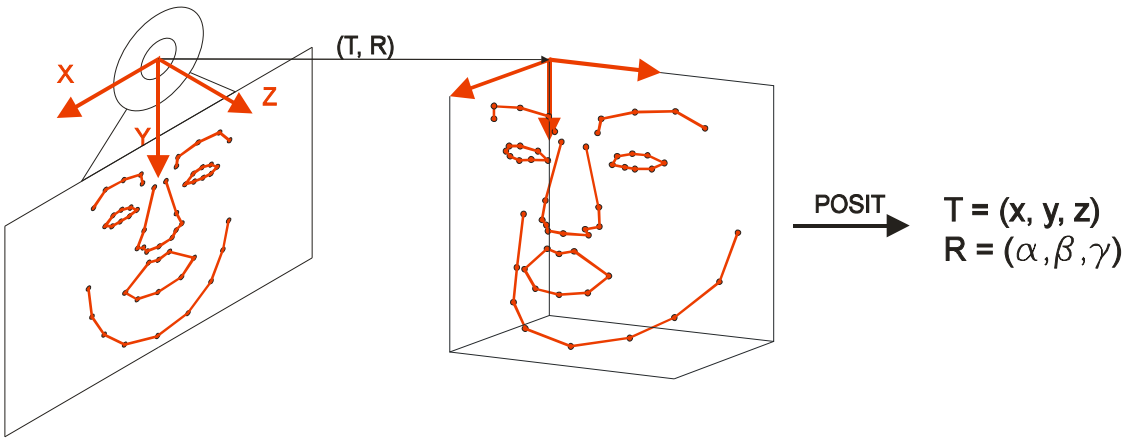


Fig. 9.2: funcionamiento de POSIT

9.3 - AAM + POSIT

AAM, *Active Appearance Models* [12], es un algoritmo con un enfoque similar a ASM, con la diferencia de que mientras ASM contempla la apariencia de la imagen en un perfil perpendicular al contorno del objeto, AAM triangula la imagen y contempla la apariencia de la región contenida en cada triángulo (Fig. 9.3).

Aplicando PCA, tanto la forma como la apariencia se pueden aproximar a partir de la forma y apariencia promedios, sus matrices de vectores propios y unos coeficientes b.

$$x \approx \bar{x} + Ps \cdot bs$$
$$g \approx \bar{g} + Pg \cdot bg$$

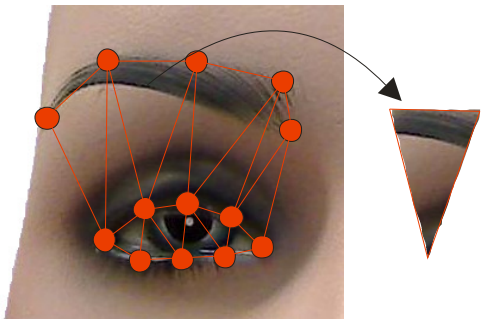


Fig. 9.3: triangulación de la imagen en AAM

Para el estudio de AAM se han utilizado 3 implementaciones diferentes, denominadas Code, Cone y Coline [13]. Todas ellas se basan en modelos de forma y apariencia, pero utilizan diferentes enfoques matemáticos para el tratamiento de esos modelos, lo que se traduce en diferentes resultados y tiempo de procesado.

Tras la segmentación mediante AAM se estima la posición de la cabeza con POSIT, del mismo modo que con ASM.

9.4 - FaceAPI

FaceAPI [15] es un sistema de estimación de la posición de la cabeza desarrollado por SeeingMachines. Aunque se trata de una aplicación comercial, existe una versión con ciertas limitaciones disponible para fines académicos. En este proyecto se utiliza la versión académica, ya que la información que proporciona es suficiente para el estudio que se pretende realizar.

Al ser un sistema comercial los desarrolladores no proporcionan información sobre el método de detección que utiliza, siendo la única información proporcionada la referente a su bajo consumo de recursos y su alta precisión (Fig. 9.4).

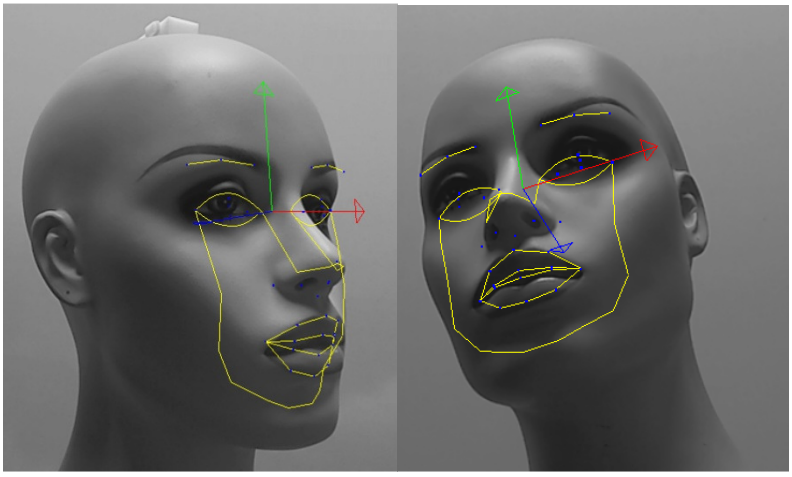


Fig. 9.4: Detección de FaceAPI

FaceAPI puede procesar imágenes sin información sobre la cámara con la que se adquieren o las puede procesar conociendo los parámetros intrínsecos de la cámara, como la distancia focal o la distorsión de la lente. Se han utilizado ambos métodos de procesado para estudiar la importancia de la calibración en los sistemas de imagen.

9.5 - Intraface

Intraface [16] es un sistema de estimación de la posición de la cabeza desarrollado por HumanSensing. Al igual que FaceAPI se trata de una aplicación comercial, pero también existe una versión disponible para fines académicos. Si bien los desarrolladores tampoco proporcionan información muy detallada sobre su funcionamiento, y el código es cerrado, el programa combina técnicas de optimización, reducción de la dimensionalidad y aprendizaje.

La principal limitación de Intraface es que no proporciona información sobre la posición de la cabeza, únicamente sobre su orientación. No obstante, a modo de comparación es interesante estudiar este método, ya que los autores aseguran una enorme precisión (Fig. 9.5).

Fig. 9.5. Estimación de Intraface

9.6 - Modelos de cabeza

POSIT requiere el modelo de tridimensional correspondiente a los puntos 2D de la imagen para estimar la posición y orientación. En este trabajo se han utilizado 4 modelos diferentes. El primero es el modelo obtenido del marcado de cada usuario, de manera que con cada usuario se estima su posición con su propio modelo de cara. El segundo es un modelo genérico. El tercero es un modelo deformable que ajusta cada órgano facial de manera independiente. El cuarto es un modelo deformable mediante PCA.

Modelo propio del usuario

El primer modelo empleado con POSIT es el modelo propio de cada usuario, modelo que se ha obtenido en el proceso de marcado de los 54 puntos faciales descrito anteriormente. El objetivo es estudiar si, como se puede pensar de antemano, el modelo que contiene la información más precisa sobre el usuario es el que mejores resultados proporciona.

Modelo genérico

Buscando un enfoque completamente diferente al anterior, el siguiente modelo empleado es un modelo de cabeza genérico. El objetivo es estudiar cuánto afecta a los resultados utilizar un modelo que no contiene ninguna información sobre la fisionomía del usuario. Partiendo del modelo de 53490 puntos desarrollado por Paysan [17], se han extraído los 54 puntos faciales de interés.

Modelo deformable por órganos

Como punto intermedio entre usar un modelo muy real pero dificilmente implementable, y usar un modelo genérico sencillamente implementable, se propone como tercer modelo un modelo deformable que se ajusta a la morfología del sujeto partiendo de un modelo genérico [18]. El método de deformación consiste en modificar la disposición y tamaño relativo de cada órgano facial de manera independiente, de manera acorde a la disposición y tamaño relativo de cada órgano en el conjunto de *landmarks* 2D (Fig. 9.6). Por ejemplo, si un usuario tiene un ojo desproporcionadamente pequeño, el modelo deformable tendrá el mismo ojo desproporcionadamente pequeño, pero proporcional al ojo real.

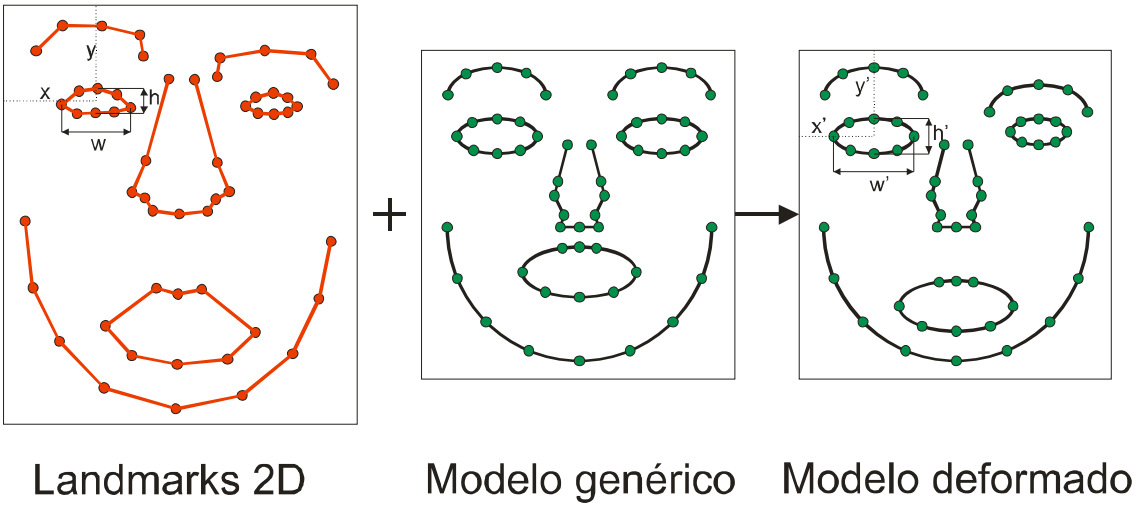


Fig. 9.6: Deformación de un modelo genérico a partir de información 2D

Modelo deformable por PCA

El último modelo propuesto consiste en un modelo deformable mediante PCA. Partiendo del modelo genérico del que se conoce su forma promedio y su matriz de vectores propios, se puede construir cualquier modelo variando el vector de coeficientes b de la siguiente manera:

$$x \approx \bar{x} + P \cdot b$$

El objetivo es encontrar el vector de coeficientes b que dé como resultado el modelo que mejor se ajusta a la cara de la persona. La principal diferencia con el método deformable por órganos es que este modelo considera la cara como una única estructura deformable, mientras que el anterior considera la cara como 7 estructuras rígidas independientes (Fig. 9.7).

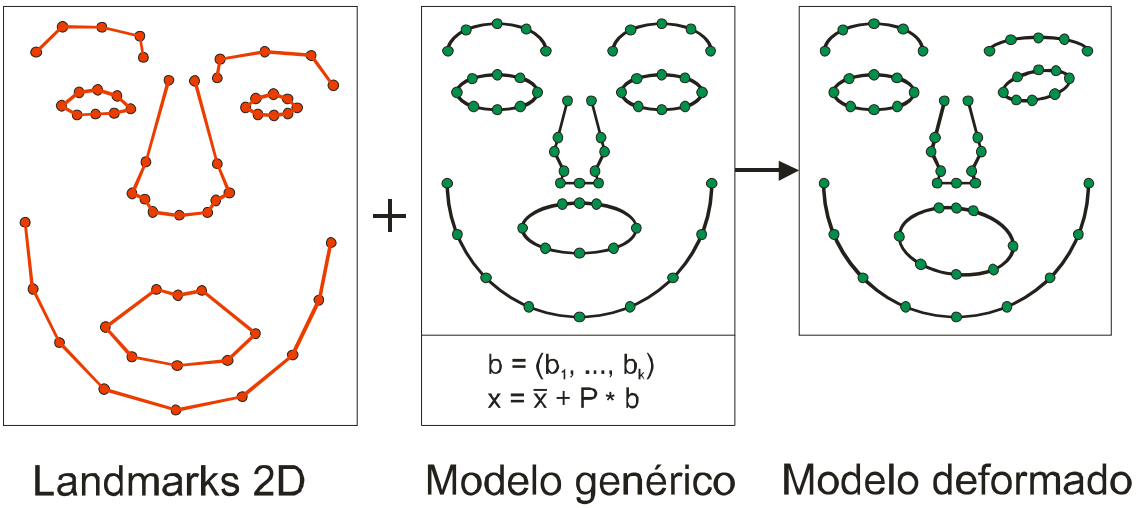


Fig. 9.7: Deformación de un modelo genérico a partir de información 2D mediante PCA

9.7 - Error sistemático debido al origen del modelo

Todo algoritmo de estimación de posición de cabeza se basa de algún modo en un modelo tridimensional. POSIT calcula la posición y orientación del modelo como la traslación y rotación, con respecto a la cámara, del sistema de referencia que contiene al modelo 3D.

Esto significa que para una posición de cabeza dada, el resultado de POSIT es diferente si se utilizan dos modelos exactamente iguales pero con diferente origen de coordenadas. En la figura 9.8, el modelo tridimensional tiene con respecto a la cámara una traslación T1 y rotación R1, si se utiliza el sistema de coordenadas 1. Del mismo modo, el modelo tiene una traslación T2 y rotación R2 si se utiliza el sistema de coordenadas 2. Pero en realidad el modelo está en la misma posición y orientación con respecto a la cámara en los dos casos.

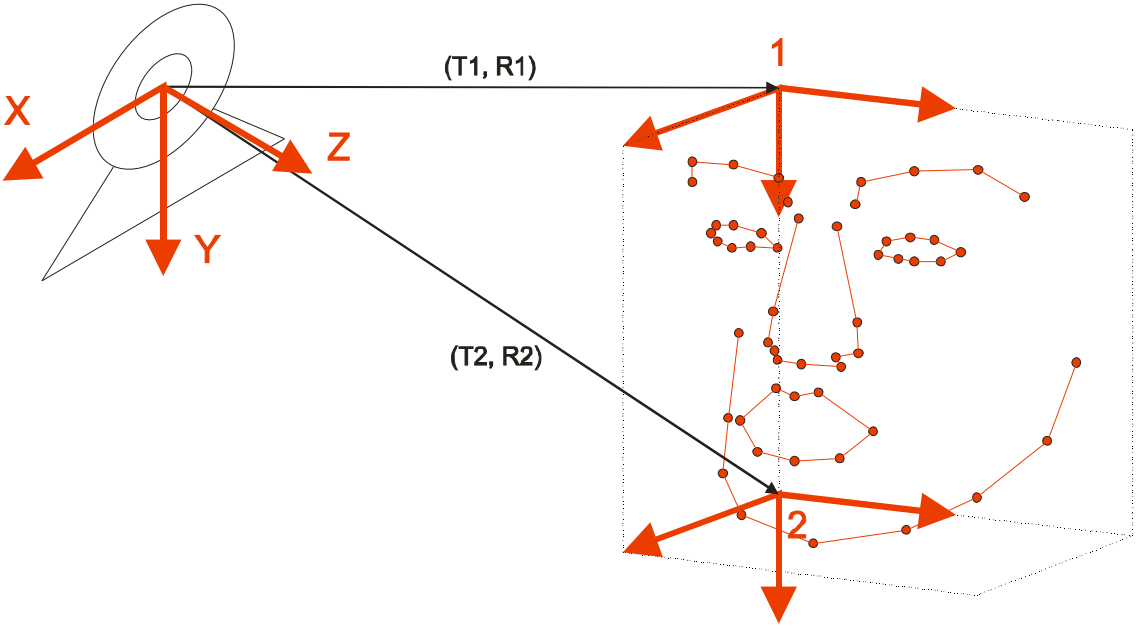


Fig. 9.8: diferentes estimaciones ante una misma posición de la cabeza

Por ello, el error en el cálculo de la posición de la cabeza tiene dos componentes, una componente aleatoria debida a los errores en los algoritmos de detección y estimación, y una componente sistemática debida a la posición del origen de coordenadas. Con el fin de hacer una comparación más justa entre los algoritmos, además de calcular el error de estimación total, se calcula el error aleatorio tras la corrección del error sistemático de cada algoritmo.

9.8 - Estabilidad del algoritmo

Al margen del error de cada algoritmo, se ha calculado la estabilidad de cada uno de ellos ante imágenes sin movimiento. Lo que se pretende medir es si, ante un usuario perfectamente estático, sin movimiento de ningún tipo, el algoritmo estima la misma posición a lo largo del tiempo o si la estimación de los datos varía, y además conocer cuánto varía. Para ello se ha generado un vídeo estático, de las mismas características que los de la base de datos, pero con un único frame que se repite a lo largo del vídeo, para asegurar una estabilidad exacta del sujeto.

En los algoritmos que segmentan la imagen (ASM, AAM e Intraface) se ha calculado la variabilidad de los *landmarks* que devuelven. En los algoritmos que calculan la traslación del sujeto (ASM, AAM y FaceAPI) se ha calculado la variabilidad de esa traslación. En los algoritmos que calculan la orientación (todos) se ha calculado la variabilidad de esa orientación.

La figura 9.9 muestra el resultado de la segmentación de ASM a una misma imagen 300 veces consecutivas. Se observa que el resultado no es el mismo en todas ellas, sino que varía notablemente. Se muestra además el resultado de la orientación estimada para la misma imagen a partir de esas 300 segmentaciones y POSIT.

Esta variabilidad viene dada porque los algoritmos utilizados tienen cierta realimentación en sus cálculos, utilizando las estimaciones anteriores como punto de partida para una nueva estimación. Es decir, no consideran cada fotograma un ente independiente, sino que está relacionado con el fotograma anterior. Esto es así porque la escena varía poco entre fotogramas consecutivos, y la complejidad del procesado se reduce bastante al limitar el área de búsqueda.

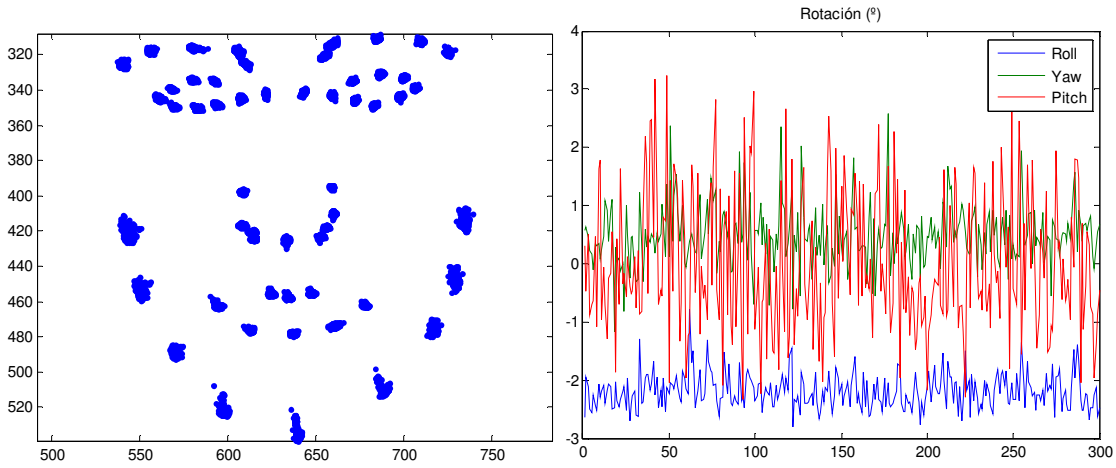


Fig. 9.9: variabilidad en la estimación de una misma imagen procesada 300 veces

10 – Resultados

En esta sección se muestran los resultados de cada método estudiado, en cuanto a error en la estimación, tiempo de procesado y estabilidad.

El error en la estimación se muestra de dos maneras diferentes. La primera es el error absoluto, calculado como la diferencia entre la estimación del algoritmo y el *ground truth*. La segunda es el error tras eliminar su componente sistemática (sección 9.7). Por ello, el error de cada método se muestra en 2 tablas, la primera que contiene el error absoluto y la segunda que contiene el error corregido, sin esa componente sistemática.

10.1 - ASM + POSIT

Error de estimación

La implementación de ASM empleada utiliza un enfoque multiresolución. Se han segmentado todas las imágenes partiendo de 5 resoluciones diferentes. Además, el resultado de cada resolución se ha procesado con POSIT utilizando los 4 modelos tridimensionales ya explicados.

Se presentan las tablas con los datos resumidos de cada método, y tras ellas se muestran tres gráficas con la información conjunta, para mayor comodidad en la interpretación de los datos.

Error promedio de rotación (º):

	MR 1	MR 2	MR 3	MR 4	MR 5
M. propio	5,04	3,02	2,60	2,52	4,73
M. genérico	5,38	3,38	3,03	2,97	5,03
M. deformable órganos	5,26	3,17	2,78	2,69	5,00
M. deformable PCA	5,38	3,38	3,03	2,98	5,03

	MR 1	MR 2	MR 3	MR 4	MR 5
M. propio corregido	4,26	2,41	2,01	1,91	3,88
M. genérico corregido	4,41	2,57	2,18	2,09	3,94
M. deformable órganos corregido	4,55	2,68	2,29	2,19	4,23
M. deformable PCA corregido	4,40	2,57	2,18	2,09	3,94

Error promedio de traslación (mm):

	MR 1	MR 2	MR 3	MR 4	MR 5
M. propio	19,86	9,84	9,05	8,86	19,42
M. genérico	73,03	66,02	65,33	65,15	70,24
M. deformable órganos	84,42	78,12	77,47	77,09	81,22
M. deformable PCA	73,35	67,23	66,95	66,58	71,19

	MR 1	MR 2	MR 3	MR 4	MR 5
M. propio corregido	15,40	7,36	6,62	6,45	14,79
M. genérico corregido	15,48	7,57	6,90	6,74	15,04
M. deformable órganos corregido	17,44	9,56	8,91	8,79	17,55
M. deformable PCA corregido	15,49	7,57	6,91	6,75	15,05

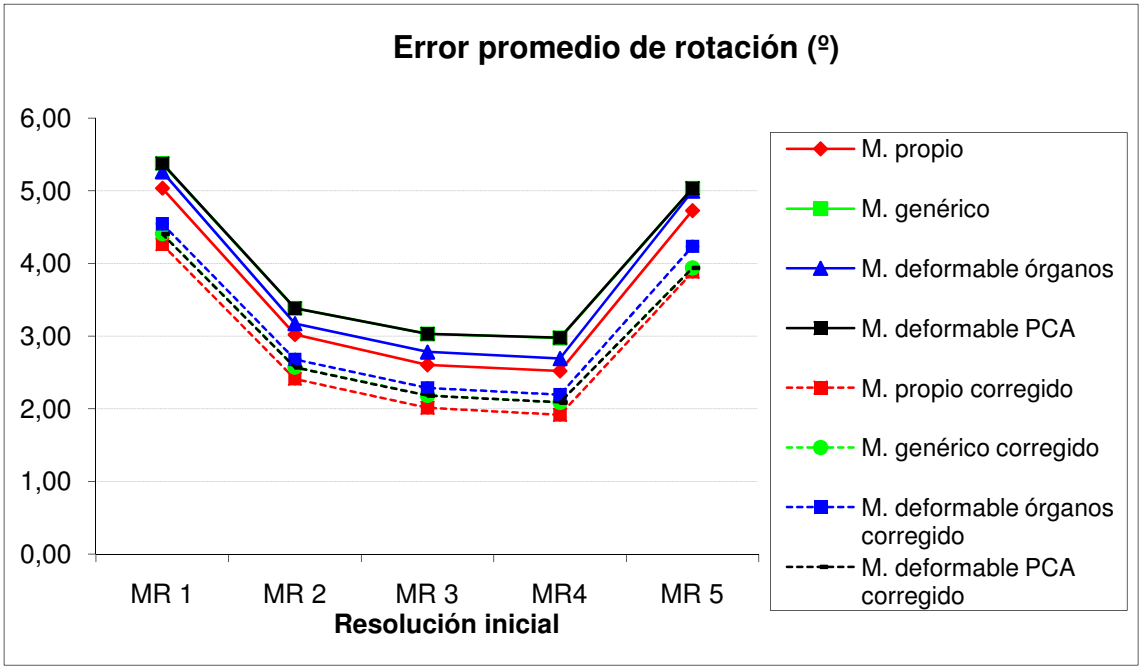


Fig. 10.1: Error promedio de rotación

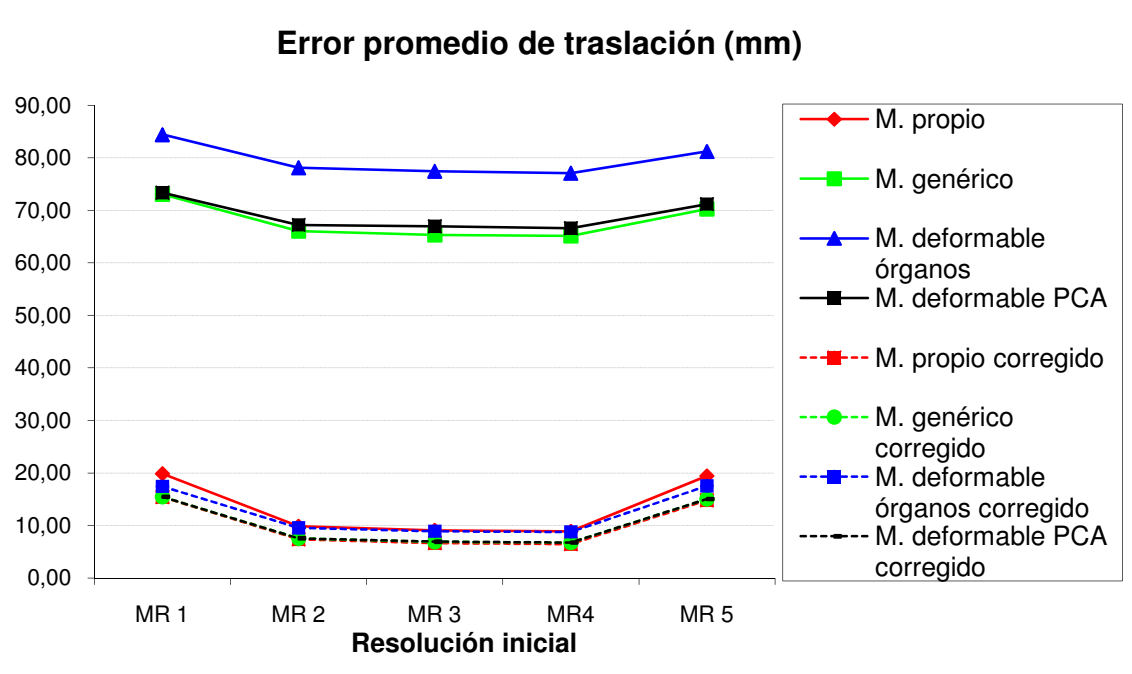


Fig. 10.2: Error promedio de traslación

Tiempo de ejecución

Se muestra el tiempo de ejecución por cada *frame* según los 5 métodos de procesado.

	MR1	MR2	MR3	MR4	MR5
tiempo (s)	2,70	4,84	6,88	7,16	8,65

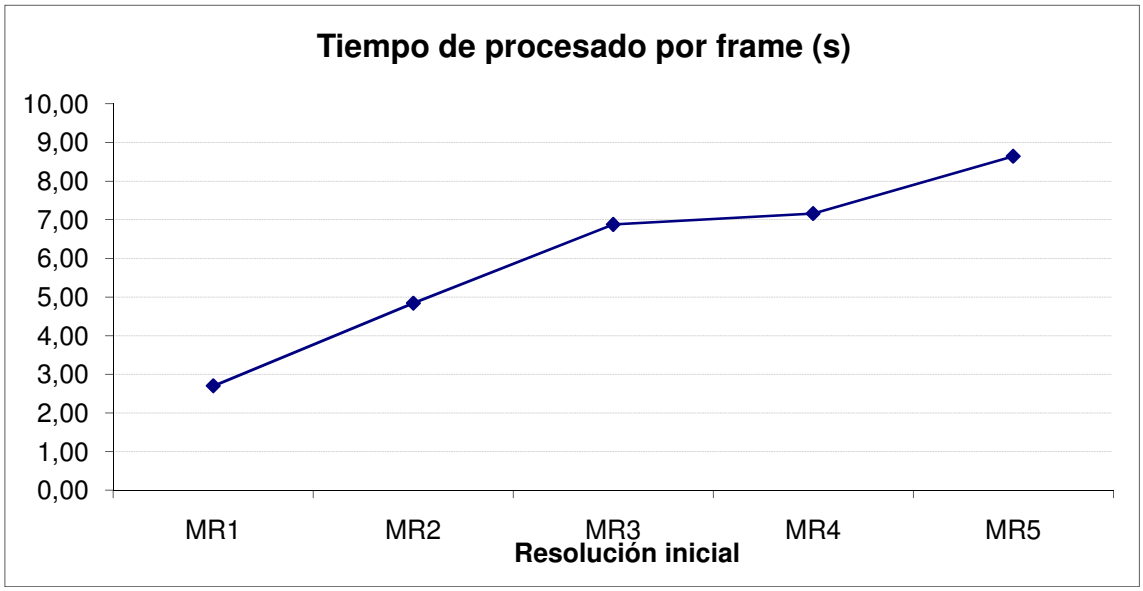


Fig. 10.3: Tiempo de procesado por imagen

Estabilidad

Se muestra la variabilidad en los resultados al procesar una misma imagen 300 veces consecutivas. Se muestra en primer lugar la variabilidad en la segmentación, como la desviación estándar de los *landmarks* estimados. Se muestra después la variabilidad en la posición y orientación de la cabeza.

	MR1	MR2	MR3	MR4	MR5
variabilidad (px)	0,75	0,83	0,92	0,94	0,94
variabilidad T(mm)	1,25	2,13	2,39	2,57	2,45
variabilidad R(º)	0,36	0,43	0,66	0,67	0,65

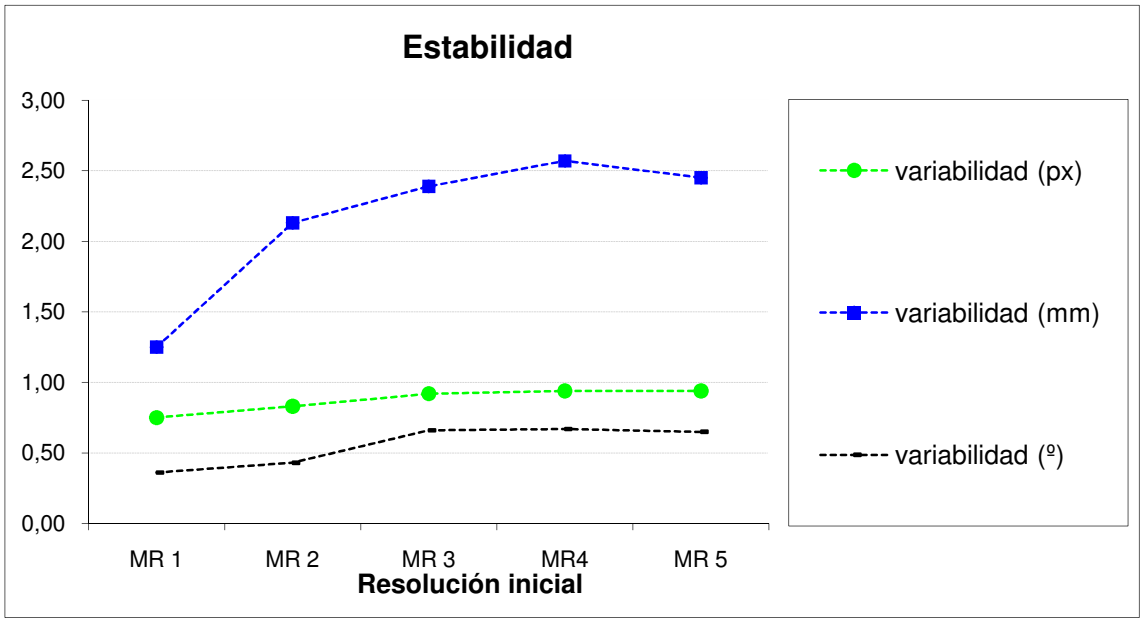


Fig. 10.4: Variabilidad de los resultados ante usuario estático

10.2 - AAM + POSIT

Error de estimación

Se muestran los resultados de las tres implementaciones de AAM: Code, Cone y Coline.

Error promedio de rotación (º):

	Code	Cone	Coline
M. propio	3,25	2,86	3,22
M. genérico	3,92	3,67	3,95
M. deformable órganos	3,52	3,12	3,41
M. deformable PCA	3,93	3,67	3,95

	Code	Cone	Coline
M. propio corregido	2,46	1,79	2,23
M. genérico corregido	2,65	1,97	2,40
M. deformable órganos corregido	2,69	2,01	2,44
M. deformable PCA corregido	2,65	1,97	2,40

Error promedio de traslación (mm):

	Code	Cone	Coline
M. propio	29,55	20,77	22,29
M. genérico	85,77	76,92	78,49
M. deformable órganos	101,01	91,11	93,28
M. deformable PCA	87,08	78,20	79,22

	Code	Cone	Coline
M. propio corregido	22,70	16,51	16,50
M. genérico corregido	22,69	17,29	16,58
M. deformable órganos corregido	25,58	20,11	19,24
M. deformable PCA corregido	22,72	17,30	16,61

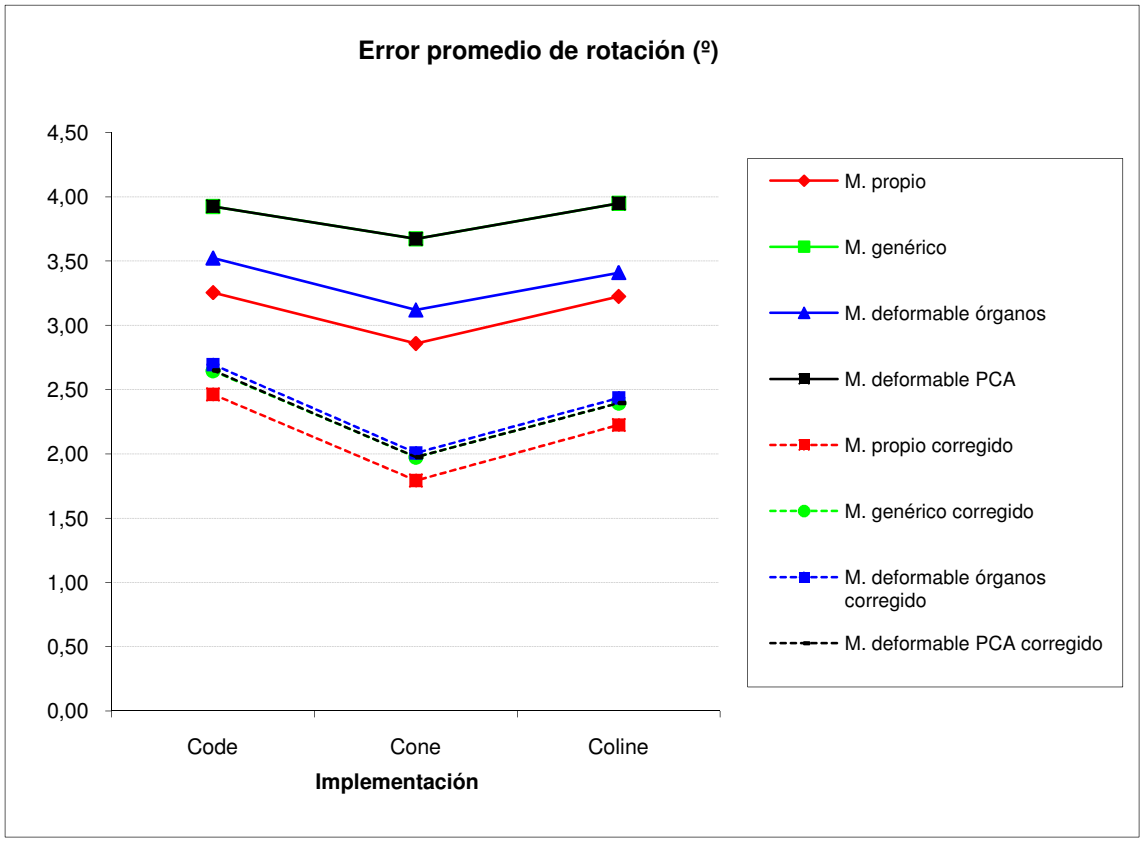


Fig. 10.5: Error promedio de rotación

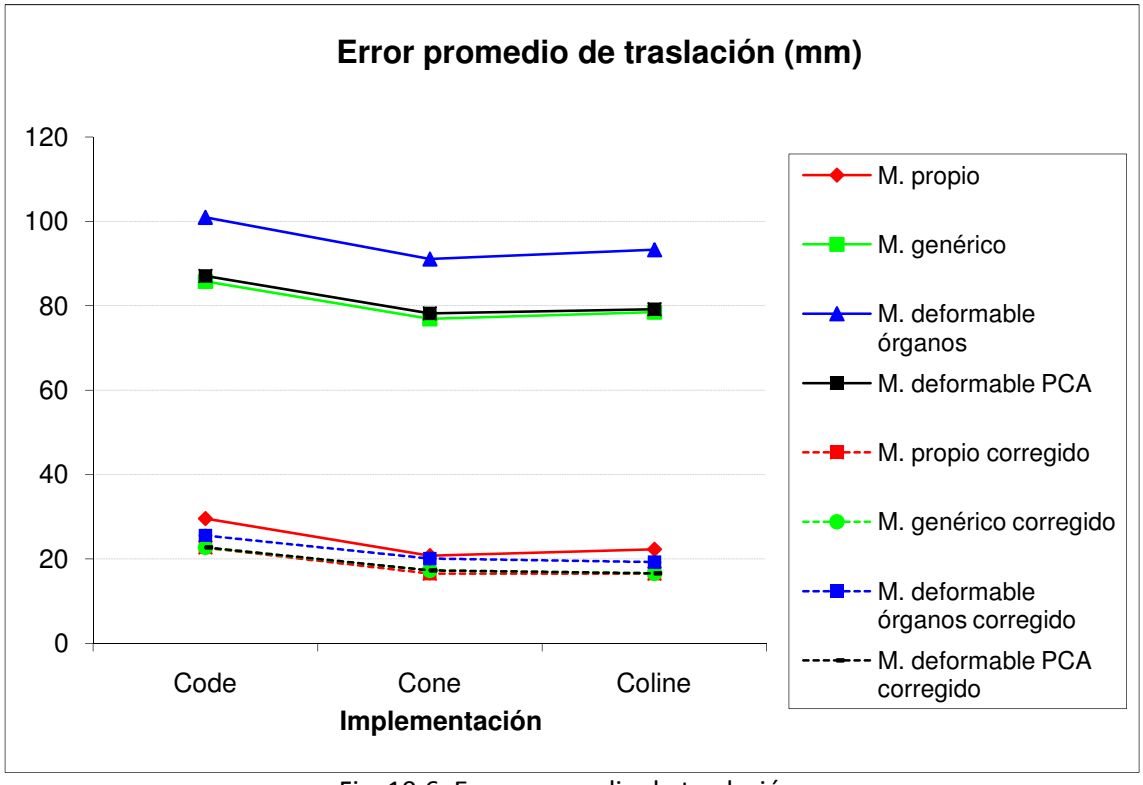


Fig. 10.6: Error promedio de traslación

Tiempo de ejecución

Se muestra el tiempo de ejecución por cada *frame* según los 3 métodos de procesado.

	Code	Cone	Coline
tiempo (s)	2,22	6,38	1,45

Estabilidad

Se muestra la variabilidad en los resultados al procesar una misma imagen 300 veces consecutivas. Se muestra en primer lugar la variabilidad en la segmentación, como la desviación estándar de los landmarks estimados. Se muestra después la variabilidad en la posición y orientación de la cabeza.

	Code	Cone	Coline
variabilidad (px)	0,81	0,47	0,65
variabilidad T(mm)	1,75	0,92	1,28
variabilidad R(º)	0,32	0,16	0,29

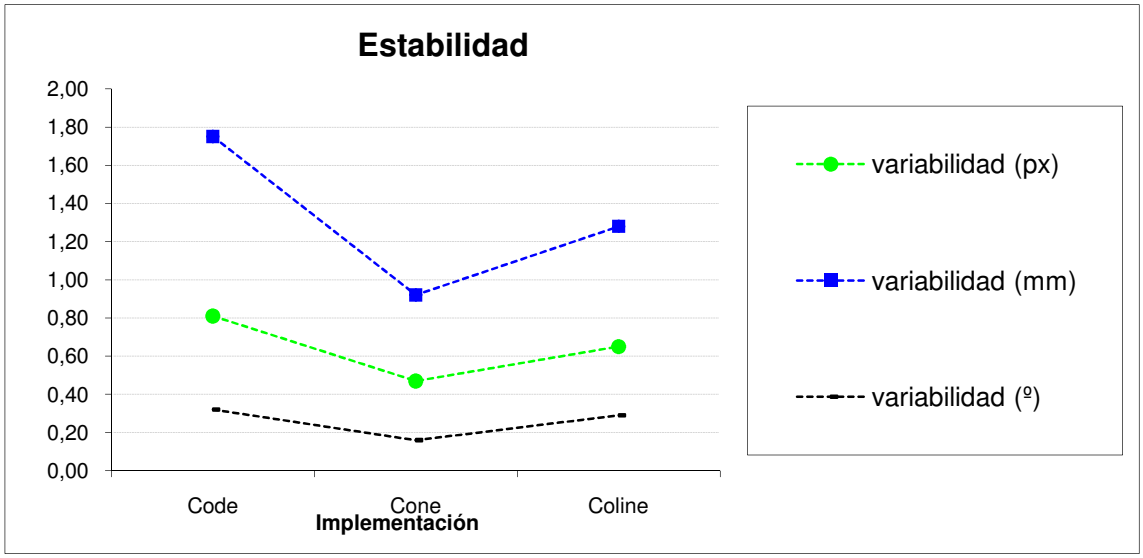


Fig. 10.7: Variabilidad de los resultados ante usuario estático

10.3 - FaceAPI

Error de estimación

Se ha comparado el resultado de FaceAPI en el procesado con información sobre los parámetros de la cámara y sin ella.

Error promedio de rotación (º):

	Sin calibración	Con calibración
error medio	5,11	4,81
error corregido	1,52	1,42

Error promedio de traslación (mm):

	Sin calibración	Con calibración
error medio	95,13	57,62
error corregido	6,31	8,48

Tiempo de ejecución

Se muestra el tiempo de ejecución por cada *frame* según los 2 métodos de procesado. Este tiempo incluye tanto la lectura como el procesado de las imágenes, mientras que en el resto de métodos se computa sólo el tiempo de procesado de las imágenes. Al ser una aplicación cerrada no es posible aislar el tiempo de procesado. No obstante, en base a estimaciones del tiempo de lectura de los datos, se cree que el tiempo de procesado es próximo al tiempo real. El tiempo de lectura depende mucho del soporte en el que se encuentras las imágenes (disco duro tradicional, disco duro SSD, cámara web), y del software empleado, de manera que lo típico en este tipo de trabajos es no contemplar ese tiempo.

	Sin calibración	Con calibración
tiempo (s)	0,12	0,12

Estabilidad

Se muestra la variabilidad en los resultados al procesar una misma imagen 300 veces consecutivas, en la posición y orientación de la cabeza.

	Sin calibración	Con calibración
variabilidad (mm)	0.0006	0,02
variabilidad (º)	0.002	0,01

10.4 - Intraface

Error de estimación

Intraface no proporciona información sobre la posición de la cabeza, únicamente sobre la orientación.

Error promedio de rotación (º):

error medio	3,49
error corregido	1,91

Tiempo de ejecución

Tiempo = 0,025 s / *frame*

Estabilidad

variabilidad (px)	0,44
variabilidad (mm)	-
variabilidad (º)	0,26

10.5 - Proyección real

Se muestran a continuación los resultados de la estimación a partir de la proyección del marcado de los *landmarks* del sujeto. Se ha calculado la estimación con los diferentes modelos, con el fin de analizar la influencia del modelo en la estimación final, supuesta una detección de los *landmarks* perfecta.

Error de estimación

Error promedio de rotación (º):

M. propio	0,02
M. genérico	2,34
M. deformable órganos	2,17
M. deformable PCA	2,34

M. propio corregido	0,02
M. genérico corregido	0,61
M. deformable órganos corregido	0,67
M. deformable PCA corregido	0,61

Error de traslación (mm):

M. propio	0,35
M. genérico	64,13
M. deformable órganos	76,10
M. deformable PCA	65,47

M. propio corregido	0,23
M. genérico corregido	2,93
M. deformable órganos corregido	5,25
M. deformable PCA corregido	2,95

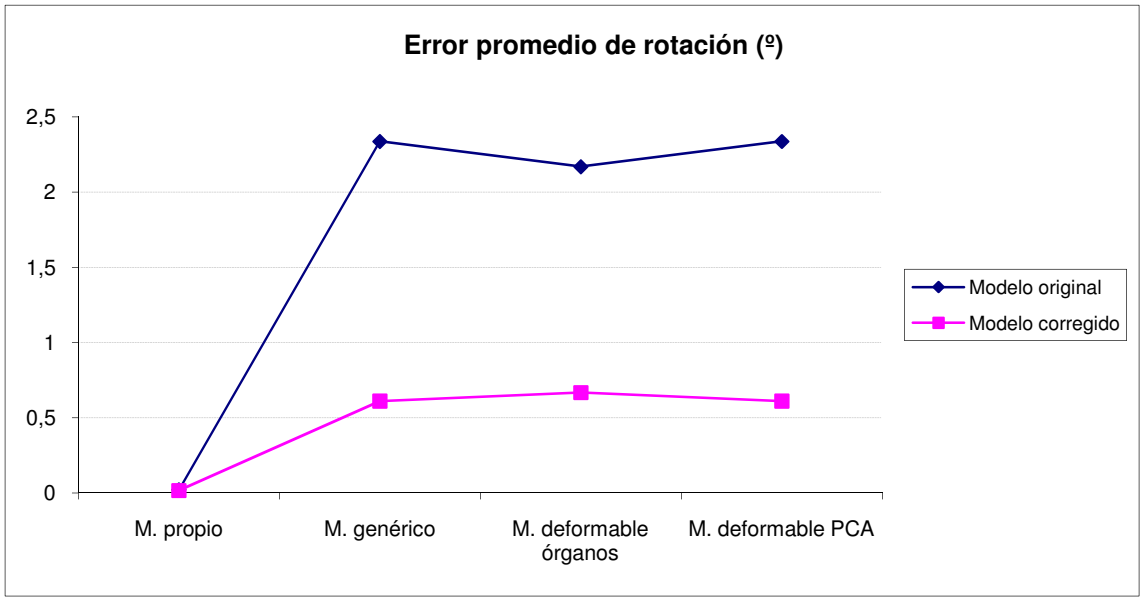


Fig. 10.8: Error promedio de rotación

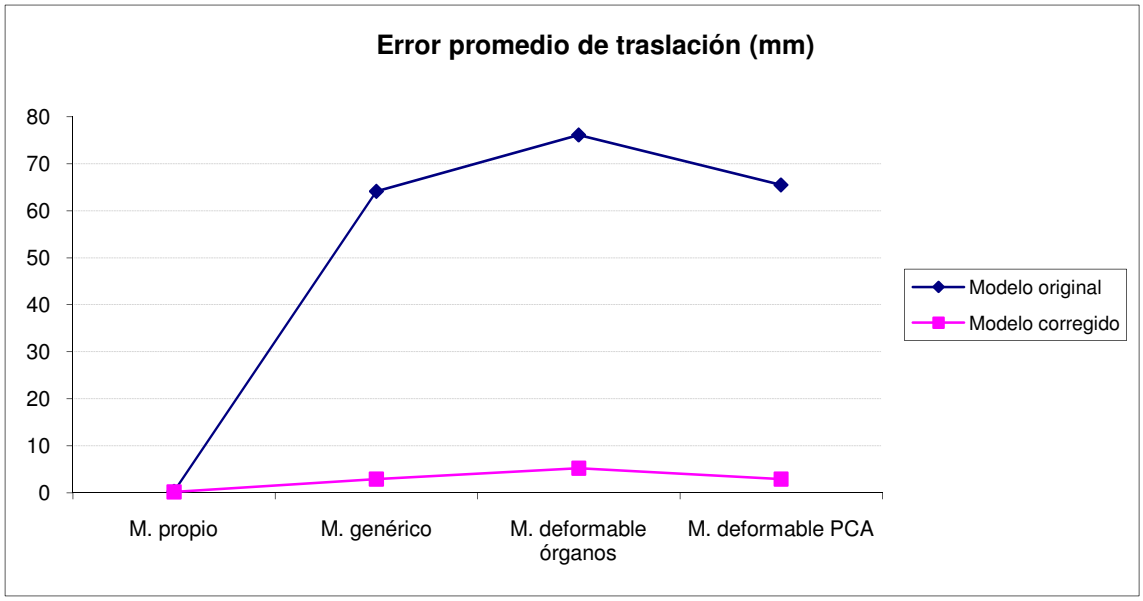


Fig. 10.9: Error promedio de traslación

Tiempo de ejecución

Tiempo de síntesis de los modelos:

	tiempo (s)
M. propio	$2 * 10^{-3}$
M. genérico	$6 * 10^{-4}$
M. deformable órganos	$2 * 10^{-3}$
M. deformable PCA	60

Tiempo de ejecución de POSIT:

Tiempo = 10^{-6} s / *frame*

10.6 - Comparativa de las mejores implementaciones de cada método

Se muestran a continuación los resultados de los 5 métodos comparados, eligiendo la mejor implementación de cada método. Se proporciona un valor único en el caso de FaceAPI e Intraface, que por su implementación utilizan su propio modelo único. Intraface no calcula la traslación del sujeto, de modo que su estimación queda vacía.

Error de rotación (º):

	ASM	AAM	FaceAPI	Intraface	Proyección real
M. propio	2,52	2,86	4,81	3,49	0,02
M. genérico	2,97	3,67	4,81	3,49	2,34
M. deformable órganos	2,69	3,12	4,81	3,49	2,17
M. deformable PCA	2,98	3,67	4,81	3,49	2,34

	ASM	AAM	FaceAPI	Intraface	Proyección real
M. propio corregido	1,91	1,79	1,42	1,91	0,02
M. genérico corregido	2,09	1,97	1,42	1,91	0,61
M. deformable órganos corregido	2,19	2,01	1,42	1,91	0,67
M. deformable PCA corregido	2,09	1,97	1,42	1,91	0,61

Error de traslación (mm):

	ASM	AAM	FaceAPI	Intraface	Proyección real
M. propio	8,86	20,77	57,62	-	0,35
M. genérico	65,15	76,92	57,62	-	64,13
M. deformable órganos	77,09	91,11	57,62	-	76,10
M. deformable PCA	66,58	78,20	57,62	-	65,47

	ASM	AAM	FaceAPI	Intraface	Proyección real
M. propio corregido	6,45	16,51	8,48	-	0,23
M. genérico corregido	6,74	17,29	8,48	-	2,93
M. deformable órganos corregido	8,79	20,11	8,48	-	5,25
M. deformable PCA corregido	6,75	17,30	8,48	-	2,95

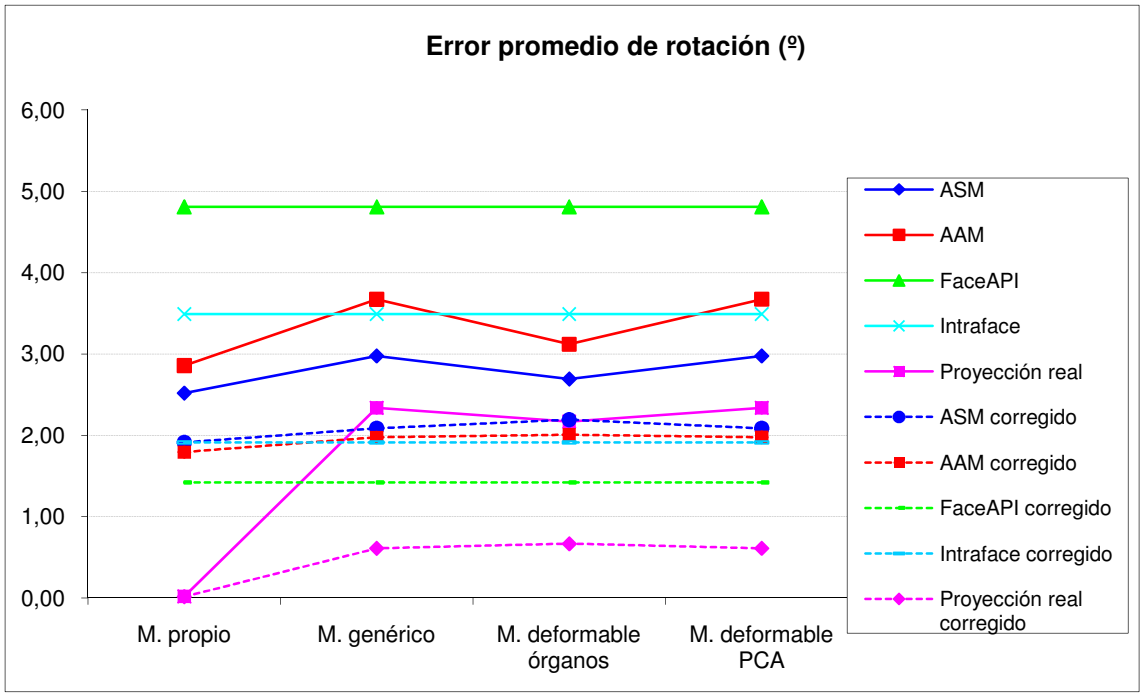


Fig. 10.10: Error promedio de rotación en los 4 métodos comparados

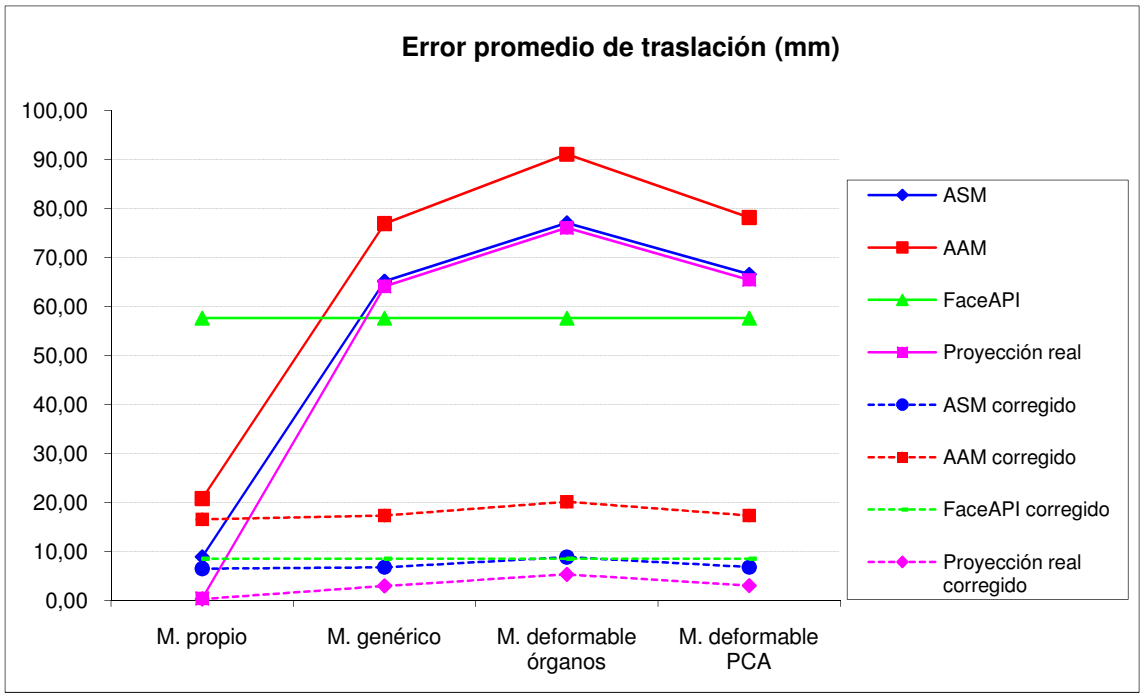


Fig. 10.11: Error promedio de traslación en los 4 métodos comparados

Se muestra además el error pormenorizado de cada método, aportando más información sobre el comportamiento de cada uno de ellos en los 6 grados de libertad. Los errores mostrados representan el mismo concepto que los anteriores, la diferencia entre la estimación y el *ground truth*, con y sin error sistemático, pero con un mayor detalle.

ASM + POSIT

	Tx (mm)	Ty (mm)	Tz (mm)	Roll (º)	Yaw (º)	Pitch (º)
M. propio	5,54	6,71	14,34	1,26	2,56	3,73
M. genérico	5,53	161,57	28,34	1,42	3,28	4,22
M. deformable órganos	8,78	171,30	51,19	1,98	2,70	3,39
M. deformable PCA	10,09	160,58	29,07	1,42	3,29	4,22

	Tx (mm)	Ty (mm)	Tz (mm)	Roll (º)	Yaw (º)	Pitch (º)
M. propio corregido	4,81	4,20	10,33	1,08	2,28	2,38
M. genérico corregido	5,52	4,21	10,47	1,19	2,58	2,49
M. deformable órganos corregido	8,81	5,10	12,46	1,72	2,28	2,59
M. deformable PCA corregido	5,52	4,23	10,50	1,19	2,58	2,48

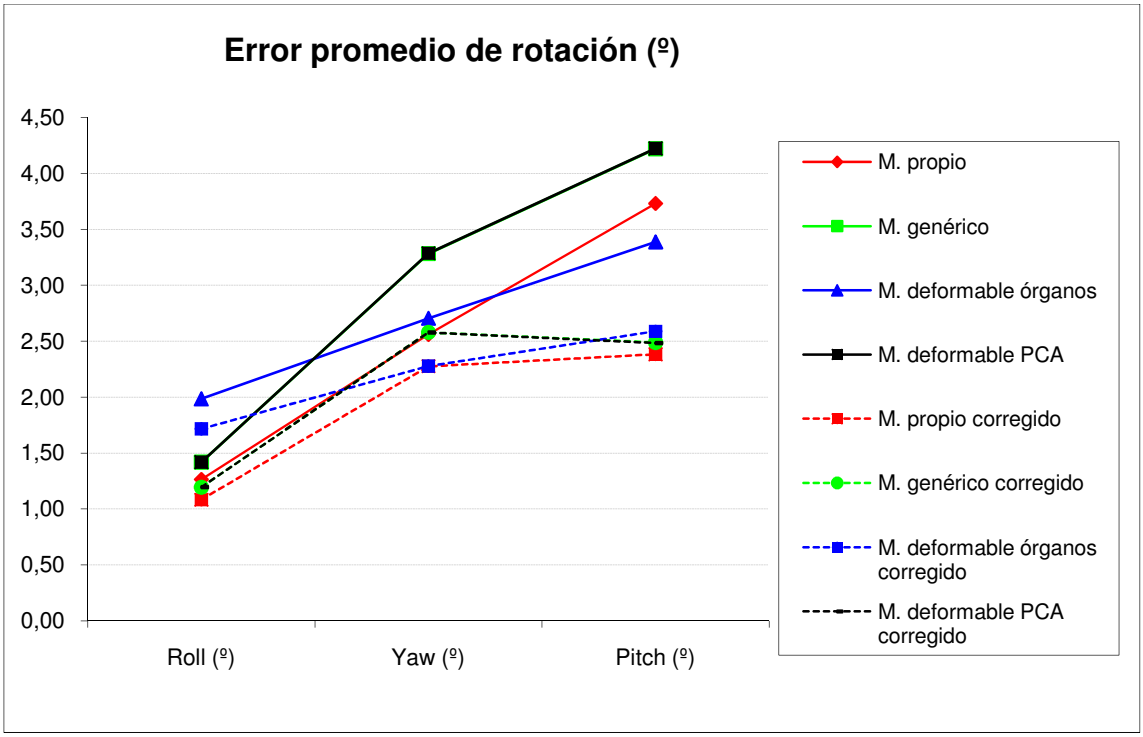


Fig. 10.12: Error promedio de rotación

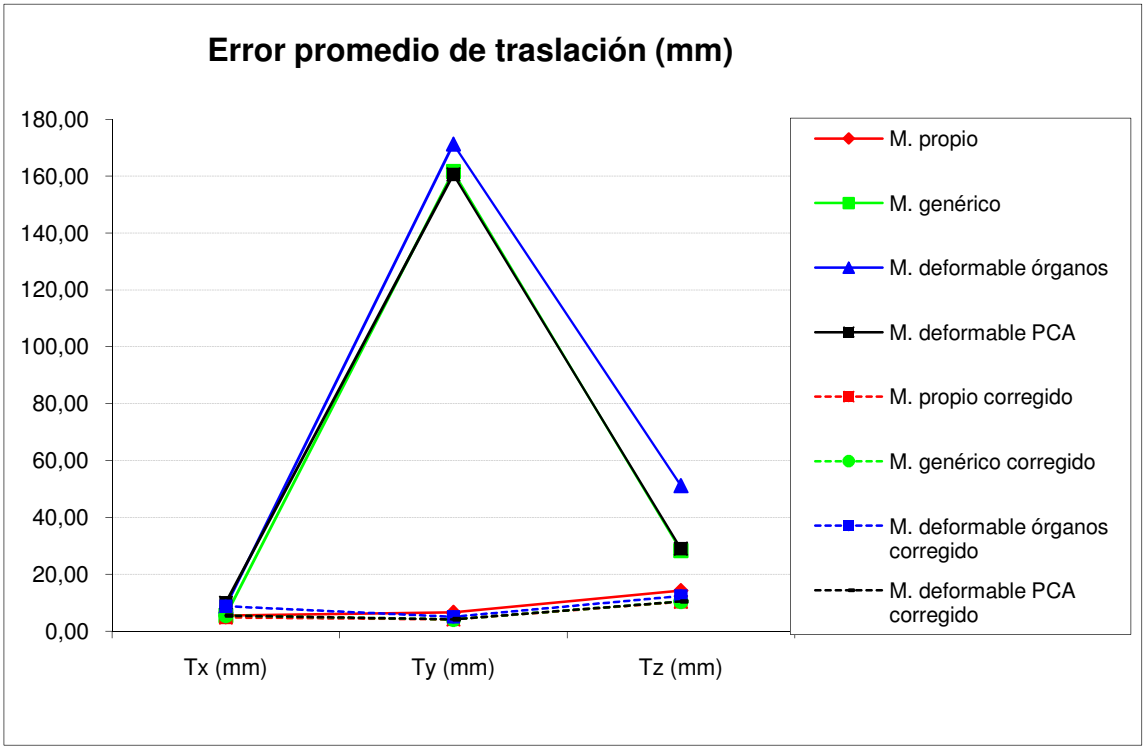


Fig. 10.13: Error promedio de traslación

AAM + POSIT

	Tx (mm)	Ty (mm)	Tz (mm)	Roll (º)	Yaw (º)	Pitch (º)
M. propio	10,60	10,42	41,29	1,98	2,23	4,36
M. genérico	10,12	162,71	57,94	1,95	2,92	6,15
M. deformable órganos	13,33	177,11	82,89	2,50	2,68	4,19
M. deformable PCA	13,77	162,13	58,69	1,96	2,92	6,14

	Tx (mm)	Ty (mm)	Tz (mm)	Roll (º)	Yaw (º)	Pitch (º)
M. propio corregido	7,76	7,39	34,38	1,45	1,83	2,11
M. genérico corregido	9,19	7,90	34,77	1,60	2,20	2,12
M. deformable órganos corregido	13,46	9,65	37,22	2,10	1,70	2,21
M. deformable PCA corregido	9,21	7,90	34,79	1,60	2,20	2,12

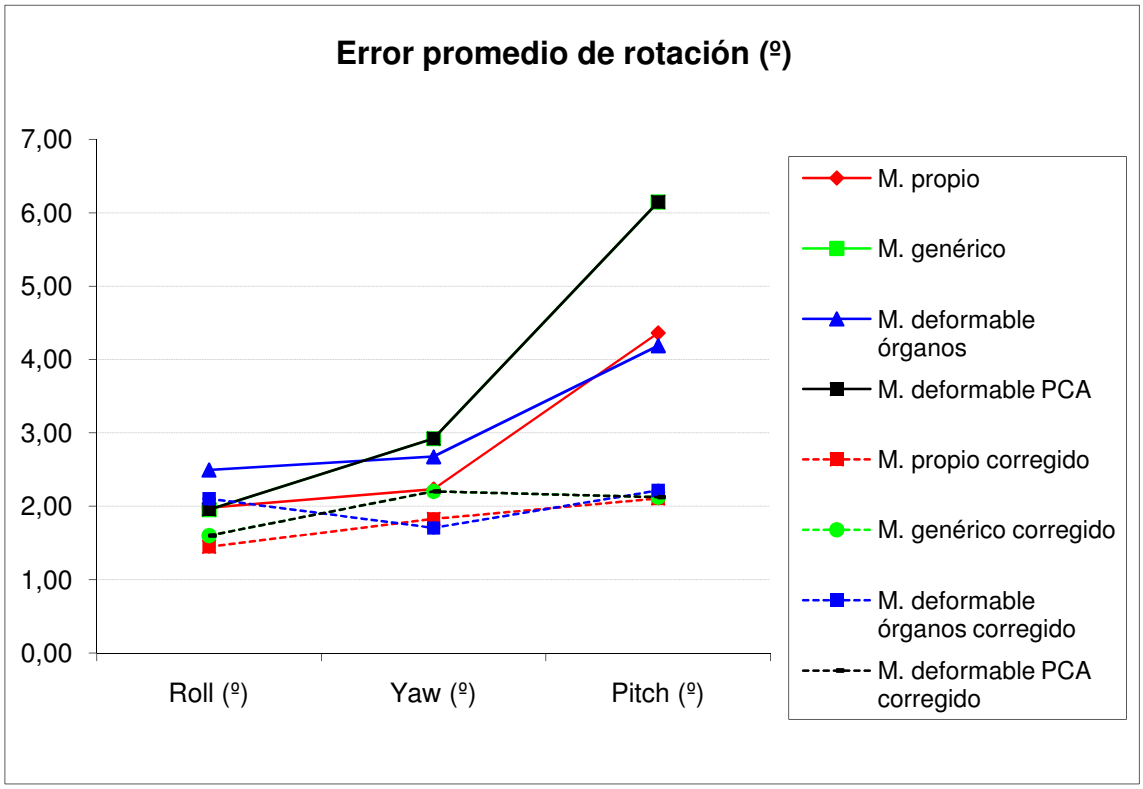


Fig. 10.14: Error promedio de rotación

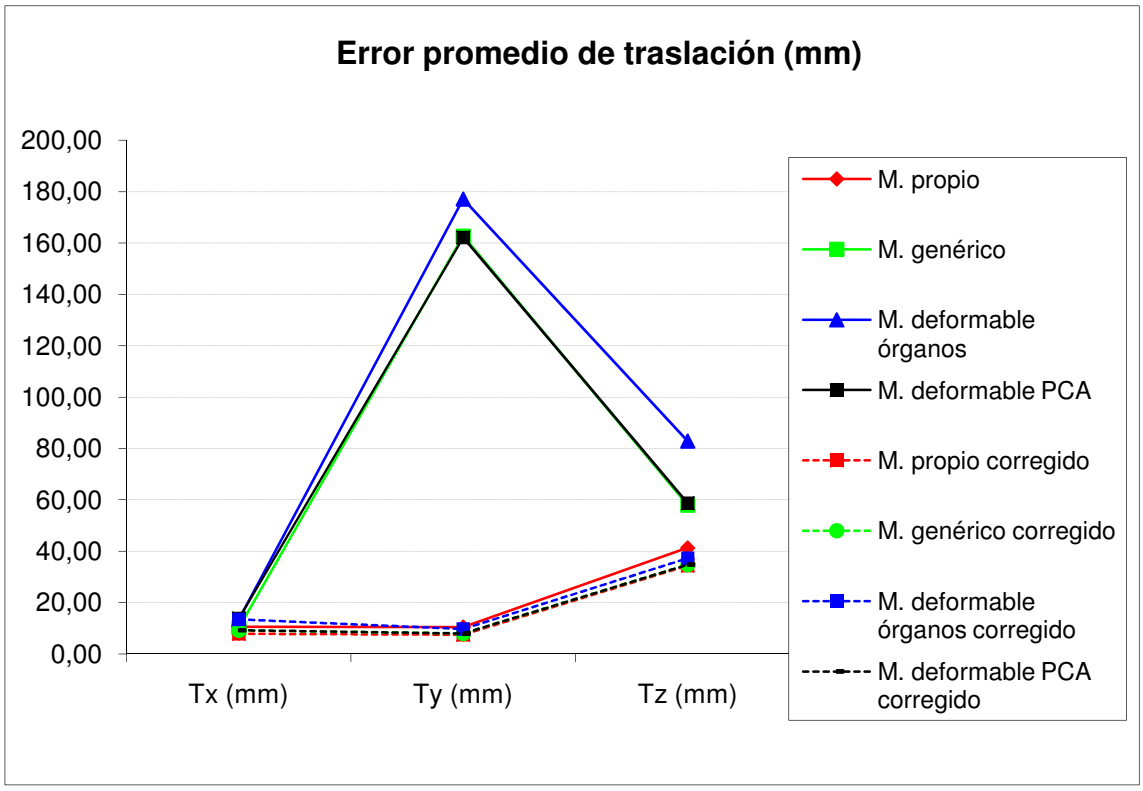


Fig. 10.15: Error promedio de traslación

FaceAPI

	Tx (mm)	Ty (mm)	Tz (mm)	Roll (º)	Yaw (º)	Pitch (º)
error medio	9,89	126,37	36,60	1,87	3,38	9,17
error corregido	9,67	6,32	9,44	0,86	1,83	1,57

Intraface

	Roll (º)	Yaw (º)	Pitch (º)
error medio	1,02	4,27	5,18
error corregido	0,78	3,60	1,36

Proyección real

	Tx (mm)	Ty (mm)	Tz (mm)	Roll (º)	Yaw (º)	Pitch (º)
M. propio	0,12	0,10	0,82	0,01	0,03	0,02
M. genérico	4,74	160,21	27,44	1,12	2,16	3,72
M. deformable órganos	7,85	173,36	47,09	1,57	1,76	3,18
M. deformable PCA	7,94	160,80	27,65	1,12	2,17	3,72

	Tx (mm)	Ty (mm)	Tz (mm)	Roll (º)	Yaw (º)	Pitch (º)
M. propio corregido	0,10	0,07	0,51	0,01	0,02	0,01
M. genérico corregido	3,62	1,78	3,39	0,64	0,70	0,48
M. deformable órganos corregido	7,14	3,01	5,61	1,19	0,42	0,40
M. deformable PCA corregido	3,61	1,78	3,46	0,64	0,70	0,49

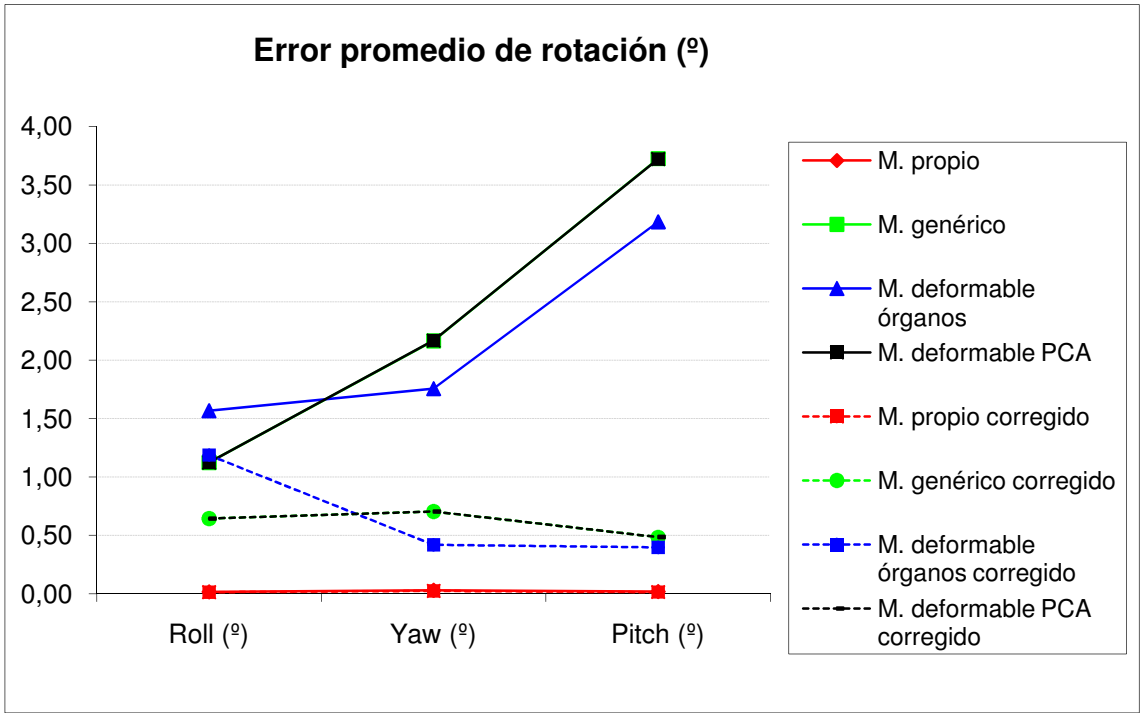


Fig. 10.16: Error promedio de rotación

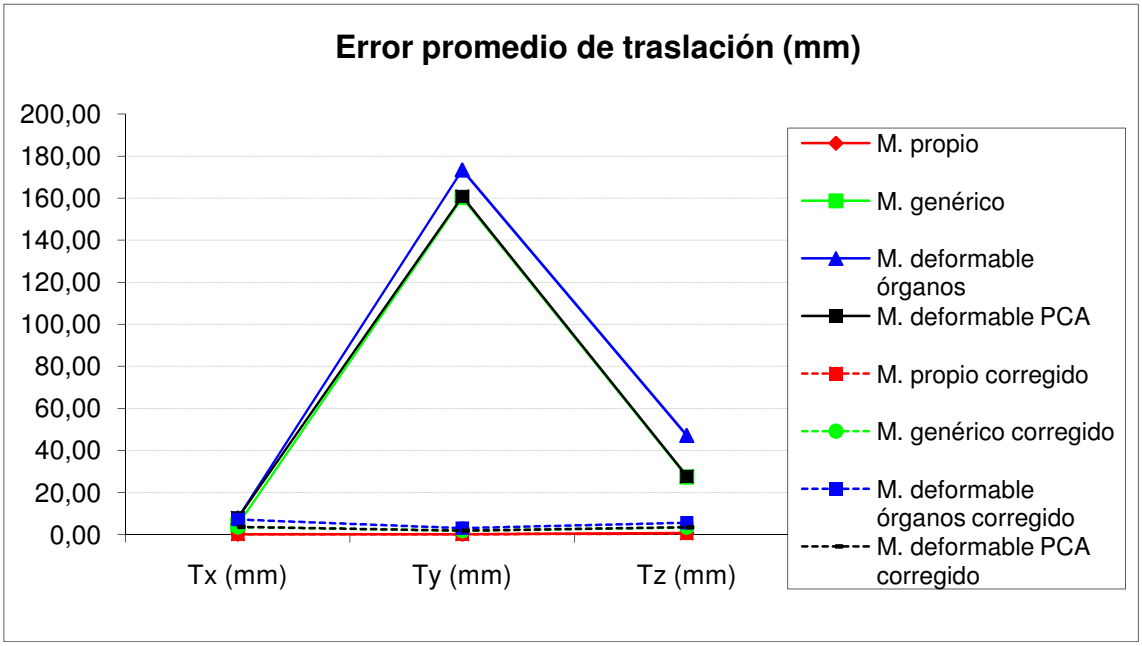


Fig. 10.17: Error promedio de traslación

Tiempo de ejecución

Se debe tener en cuenta que el tiempo de FaceAPI incluye no sólo el procesado sino la lectura de las imágenes, de modo que el tiempo real es algo menor al mostrado.

	ASM	AAM	FaceAPI	Intraface	Proyección real
tiempo (s)	7,16	6,38	0,12	0,02	0,000001

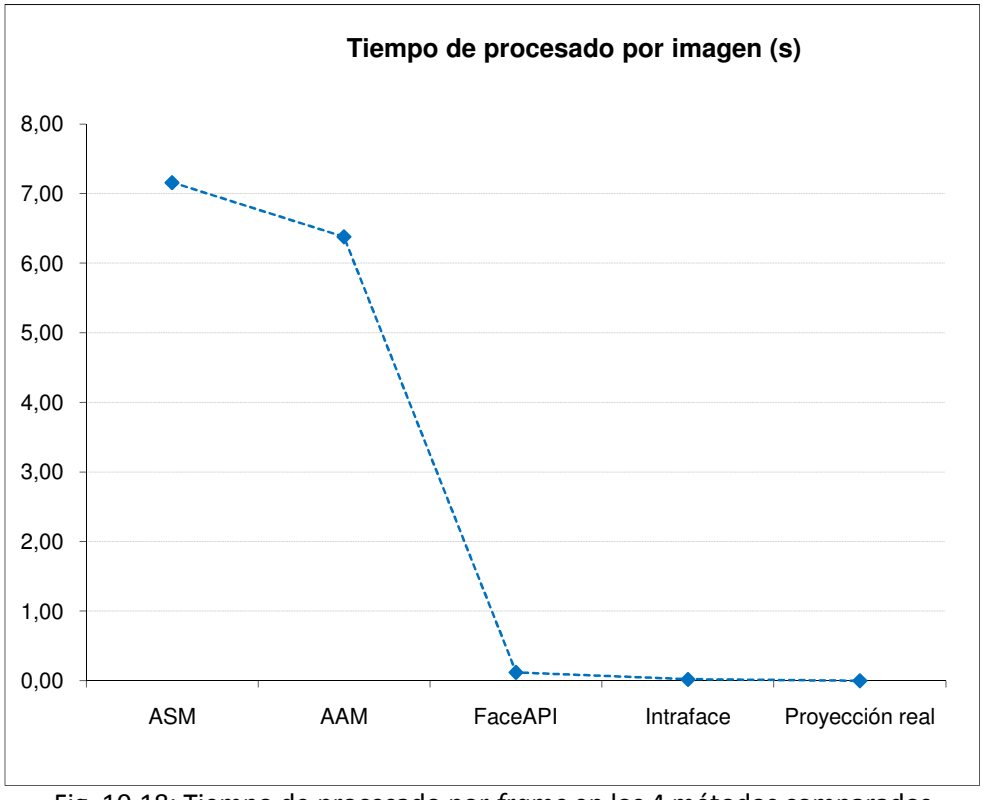


Fig. 10.18: Tiempo de procesado por *frame* en los 4 métodos comparados

Estabilidad

Se muestra la variabilidad en los resultados al procesar una misma imagen 300 veces consecutivas. Se muestra en primer lugar la variabilidad en la segmentación, como la desviación estándar de los landmarks estimados. Se muestra después la variabilidad en la posición y orientación de la cabeza.

	ASM	AAM	FaceAPI	Intraface
variabilidad (px)	0,94	0,47	-	0,44
variabilidad (mm)	2,57	0,92	0,02	-
variabilidad (º)	0,67	0,16	0,01	0,26

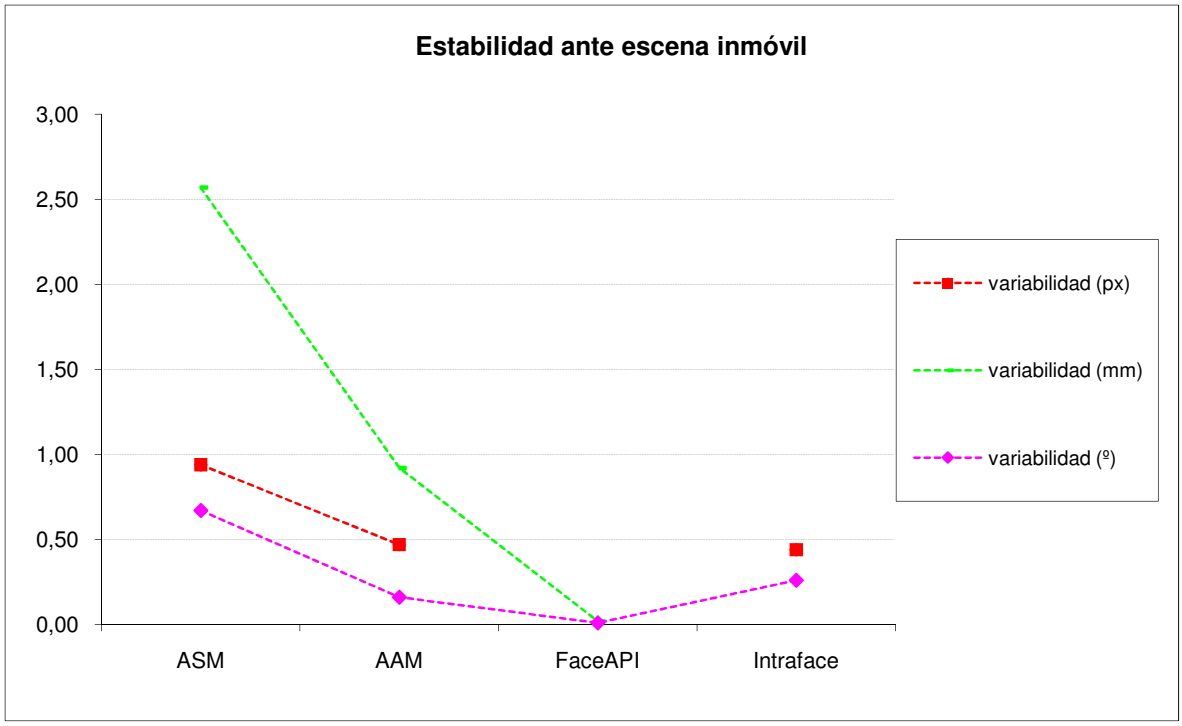


Fig. 10.19: Estabilidad en los 4 métodos comparados

11 – Análisis de los resultados

11.1 - Calibración

Tras los cambios realizados en el proceso de calibración, tanto la toma de datos como el proceso de calibración en sí son dos procesos significativamente más rápidos que la versión previa, y el esfuerzo por parte de la persona que realiza la calibración es considerablemente menor.

La automatización de los procesos supone reducir el tiempo de calibración de entre 2 y 3 horas a 30 minutos, y la optimización de los procesos supone reducir el error de proyección de 6.2 a 1.5 pixeles. Sin lugar a dudas queda patente el éxito de las modificaciones realizadas.

El número de imágenes para la calibración influye en el cálculo de los parámetros intrínsecos de la cámara. Una calibración con menos de 10 imágenes resulta nefasta, y sólo a partir de 30 imágenes los parámetros comienzan a estabilizarse. En torno a 50 imágenes se considera que se alcanza el equilibro entre la estabilidad de los parámetros y el tiempo de calibración.

11.2 - Marcado automático

El sistema de marcado automático ha supuesto un reto tan difícil como interesante. Para poder aceptar los resultados de este sistema ha sido necesario actuar en todas y cada una de las etapas implicadas en el proceso, desde la captura de datos del sensor hasta los parámetros de proyección de la cámara. Tras optimizar el sistema, el resultado es perfectamente útil, y ha evitado un año entero de marcado manual de las imágenes.

Se estima que el error promedio de este método es de 0.7 mm, un error totalmente asumible. Dado que durante las grabaciones se combinan dos marcados, este error puede incluso reducirse. Se puede concluir pues que este sistema es tan preciso como el marcado manual pero significativamente más rápido y menos tedioso.

11.3 - Estimación de la posición de la cabeza

11.3.1 - ASM + POSIT

Error de estimación

El primer parámetro a estudiar es la resolución de inicio en el algoritmo ASM. Se desea conocer si el error en la estimación depende del nivel de resolución inicial o no. De acuerdo a la figura 10.1, el error en la rotación es bastante diferente en las resoluciones 1 y 5 frente a las resoluciones 2, 3 y 4. Para conocer cómo de diferentes son los resultados, o si realmente lo son, se realiza un test de análisis de la varianza (ANOVA), estudiando el error de estimación frente a la resolución.

El test de ANOVA, con un nivel de significación del 5%, indica que los resultados de las resoluciones 1 y 5 son significativamente diferentes al resto (p-valor = 0). La resolución 2 proporciona un error significativamente menor a las resoluciones 1 y 5, y las resoluciones 3 y 4 proporcionan unos errores semejantes entre sí pero menores que el resto de resoluciones. Con una estimación puntual del error ligeramente menor, se opta por la elección de la resolución 4 como la más precisa.

El test de ANOVA sobre la influencia del modelo tridimensional empleado con POSIT indica que el peor resultado se obtiene utilizando el modelo genérico y el modelo deformable mediante PCA. El modelo deformable por órganos proporciona un menor error que los anteriores, pero mayor que el modelo propio del usuario. Es decir, el modelo con el que mejores resultados se obtienen es el modelo específico de cada usuario.

El error de traslación absoluto es estadísticamente similar en las 5 resoluciones (p-valor = 0,991). No obstante, tras la eliminación del error sistemático sí se encuentran diferencias significativas (p-valor = 0,000). Las resoluciones 2, 3 y 4 arrojan el menor error, error similar entre ellas 3.

En cuanto a los diferentes modelos, el mayor error se obtiene con el modelo deformable por órganos, seguido por los modelos genérico y deformable por PCA, y el menor error se obtiene con el modelo propio. Tras la corrección del error sistemático no se observan diferencias significativas entre los diferentes modelos (p-valor = 0,832).

En base a los resultados, se decide que la comparación con los otros métodos se debe realizar con la resolución inicial 4, porque estadísticamente es el algoritmo con menor error.

Tiempo de ejecución

El tiempo de procesado aumenta conforme aumenta el número de resoluciones segmentadas. Los tiempos de ejecución son considerables, casi 3 segundos por *frame* en el mejor de los casos, y hasta 7 segundos en la resolución 4, la que proporciona menor error. No es una implementación que se pueda utilizar en tiempo real.

Estabilidad

A pesar de que la segmentación comenzando en la resolución 4 es la que menor error promedio arroja, lo curioso es que la variabilidad de los resultados ante una escena estática es mayor que en el resto, en cuanto a segmentación y posición con POSIT. No obstante, una variabilidad de 2.5 mm y 0.7º puede ser aceptable.

11.3.2 - AAM + POSIT

Error de estimación

El análisis de ANOVA indica que el error varía significativamente en función del método de AAM utilizado (p-valor = 0), el modelo de cabeza para POSIT (p-valor = 0,001) y la corrección del error sistemático (p-valor = 0).

El error promedio de rotación es significativamente menor en el método Cone, hecho que se acentúa al observar el error tras la corrección de la componente sistemática. En cuanto al error de estimación de la traslación, no existen diferencias significativas entre los tres métodos. Esto se debe a que la rotación se estima a partir de la disposición relativa de los *landmarks*, y la traslación (principalmente en el eje z, en la profundidad) está más ligada al tamaño de los *landmarks*. Por ello, segmentaciones con resultados semejantes en cuanto a tamaño pueden suponer resultados completamente diferentes en cuanto a rotación pero similares en cuanto a traslación.

En cuanto al modelo tridimensional, el test de ANOVA muestra que los modelos genérico y deformable mediante PCA proporcionan un error de rotación semejante, significativamente peor que el resto. El modelo propio arroja unos errores comparables al modelo deformable por órganos. Tras la corrección del error sistemático, no existen diferencias relevantes entre los 4 modelos (p-valor = 0,861).

Resultado diferente se obtiene al analizar el error en la traslación. En este caso el modelo propio proporciona el menor error. Los modelos genérico y deformable por PCA son estadísticamente semejantes, y el modelo deformable por órganos supone un error significativamente mayor.

En base a estos resultados se decide que la comparación con los otros métodos se debe realizar con el algoritmo Cone, porque estadísticamente es el algoritmo con menor error.

Tiempo de ejecución

El algoritmo que mejores resultados muestra es a su vez el que más tiempo de ejecución necesita para procesar una imagen. Utiliza una aproximación matemática más compleja que los otros métodos, de ahí esa diferencia de tiempo. Con un procesado por *frame* de 1.5s en el mejor de los casos, y 6.4s en el peor, esta implementación de AAM no se puede utilizar para aplicaciones en tiempo real.

Estabilidad

El algoritmo Cone, además de ser el más preciso, es el más estable de los 3 comparados. Con una variabilidad inferior a 0.2º en la rotación e inferior a 1 mm en la traslación se concluye que es un método muy estable ante escenas estáticas, con una variabilidad casi despreciable.

11.3.3 - FaceAPI

Error de estimación

El error estimado para la rotación (5.1º) se reduce ligeramente cuando la aplicación conoce los parámetros de la cámara (4.8º) con la que se ha tomado el vídeo, pero esa diferencia no es estadísticamente significativa.

En el caso del error de traslación lo observado es completamente diferente, se reduce en un 40% cuando se conocen los parámetros de la cámara. Esta diferencia se observa principalmente en el cálculo de la posición en el eje z, la distancia con respecto a la cámara.

Tras eliminar el error sistemático se obtiene un error medio de rotación de 1.5º, y un error de traslación inferior a 10mm, un error considerablemente bueno, considerando que la grabación se realiza a una distancia media de 60 cm de la cámara.

En base a estos resultados se decide que la comparación con los otros métodos se debe realizar con el sistema calibrado, porque estadísticamente es el algoritmo con menor error.

Tiempo de ejecución

El tiempo medio de procesado de una imagen es de 0.12 segundos, tanto en el procesado con calibración como en el procesado sin ella. Es una aplicación muy rápida, que si bien no se puede considerar tiempo real (no llega a 10 imágenes por segundo), es uno de los métodos más rápidos estudiados. No obstante, este tiempo incluye la lectura de los datos, un proceso que incrementa el tiempo total. Tras una estimación algo grosera de ese tiempo, se puede concluir que este sistema puede trabajar en tiempo real.

Estabilidad

La característica que más sorprende de este algoritmo, al margen de su velocidad, es su gran estabilidad ante escenas no cambiantes. La variabilidad en la traslación ante una imagen fija es de 0.02 mm, y 0.01º en la rotación. Son unos datos realmente buenos, hasta tal punto que se pueden considerar despreciables. Curiosamente cuando se utiliza el algoritmo sin calibrar la estabilidad es incluso mejor, obteniendo una variabilidad de un orden de magnitud inferior. No se encuentra una explicación evidente para esta reducción de la variabilidad.

11.3.4 - Intraface

Error de estimación

El error promedio en la estimación de la rotación es de 3.5º, algo elevado. No obstante, al eliminar la componente sistemática el error se reduce a 1.9º. Es un error bajo, pero no es demasiado bueno siendo una herramienta comercial. Además, el hecho de no estimar la posición del usuario lo convierte en un método bastante limitado.

Tiempo de ejecución

El mejor aspecto de este programa es sin lugar a dudas su tiempo de ejecución, de 0.025s por *frame*. Esto permite un procesado en tiempo real, muy interesante por ejemplo en aplicaciones en tiempo real como la estimulación magnética transcraneal. Por otro lado, dado que además de la rotación proporciona un conjunto de *landmarks*, se puede combinar con otros algoritmos que requieran algún tipo de inicialización. Por ejemplo, ASM y AAM necesitan información acerca de la localización de la cara antes de comenzar la segmentación, para acotar de algún modo la zona de búsqueda. Intraface puede resultar una ayuda muy interesante para automatizar esa inicialización, ya que el incremento del tiempo de procesado es despreciable frente al tiempo propio de ASM y AAM.

Estabilidad

La variabilidad de Intraface ante una escena estática es de 0.26º. Es un error muy interesante,

11.3.5 – Marcado real

Error de estimación

Resulta interesante estudiar el resultado de la estimación cuando se utiliza POSIT con los puntos 2D del marcado facial, con el objetivo de cuantificar el error debido a POSIT con los diferentes modelos, como si se tratase de una segmentación ideal.

El error al utilizar el modelo tridimensional real es de 0.02º y 0.35 mm. Es un error realmente bueno como estimación, pero muestra que el algoritmo POSIT no es ideal. El hecho de aproximar el proceso de proyección para simplificar el procesado se traduce en la aparición de este error, aunque también en un tiempo de procesado irrisorio.

El error al utilizar cualquiera de los 3 modelos restantes se incrementa con respecto al error obtenido con el modelo propio, algo que era de esperar dado que el mejor modelo a utilizar con las marcas reales de cada persona es el propio modelo de cada persona.

No existen diferencias significativas en el error en la rotación estimada al utilizar el modelo genérico y los dos deformables, tanto al contemplar el error total como al eliminar la componente aleatoria del error. Es un hecho un tanto curioso, ya que el objetivo de los modelos deformables es caracterizar mejor la cara que el modelo genérico. Sí que es cierto que el error medio con el modelo deformable por órganos es menor, y en los métodos

anteriores resulta exitoso, pero con una segmentación perfecta se observa que la diferencia no es relevante.

En cuanto a la traslación se obtiene un error superior a 60 mm al utilizar los 3 modelos anteriores, pero se reduce a menos de 5 mm al eliminar la componente sistemática, sin duda un error muy bueno.

Tiempo de ejecución

El tiempo medio que tarda POSIT en estimar la localización y orientación del usuario es del orden del microsegundo. Por ello, cumple ampliamente con los requisitos de una aplicación para tiempo real. Además, el hecho de introducir un error tan bajo supone que los errores y latencias en la estimación van a estar determinados por el resto de etapas.

La síntesis del modelo tridimensional es otro aspecto a valorar de cara a su utilización en tiempo real. El tiempo necesario para sintetizar el modelo real, el genérico y el deformable por órganos es del orden de milisegundos. El modelo deformable por PCA supone un tiempo de 60 segundos, totalmente inviable para una aplicación *online*. Dado que los errores obtenidos con el modelo deformable mediante PCA son siempre semejantes a los obtenidos con el modelo genérico, se descarta el modelo deformable por PCA para este tipo de aplicaciones.

De acuerdo a estos datos, se considera que la etapa que más retardo introduce en la estimación de la posición de la cabeza es la segmentación, ya que una vez segmentada la imagen el resto del proceso no supone más de 2 milisegundos.

11.3.6 - Comparación de algoritmos

Error de estimación

Tras elegir la mejor implementación de cada método de estimación de la posición de la cabeza, se ha realizado un test de ANOVA para analizar las diferencias y similitudes entre ellos. Con un p-valor igual a 0, se observa que al menos uno de los métodos tiene un error significativamente diferente al resto.

FaceAPI es el método que proporciona el mayor error absoluto de la rotación, con un valor medio de 4.8º. Menor es el error de Intraface y AAM, que tienen un comportamiento estadísticamente similar entre sí, con un error medio de 3.5º y 3.3º respectivamente. ASM tiene un error menor que los anteriores, con un promedio de 2.8º. Finalmente, el método que utiliza la proyección de los puntos marcados es el que mejores resultados proporciona, con un error promedio de 1.7º.

Tras la corrección del error sistemático, se observa que las estimaciones mediante ASM, AAM e Intraface son similares, con un error de 2º. FaceAPI, el método con el mayor error absoluto, tiene un error aleatorio significativamente menor que los métodos anteriores, con un promedio de 1.4º. Por último, usar la proyección real supone un error inferior a 0.5º.

El test de ANOVA sobre el error de traslación muestra que AAM es el método de segmentación que peor estimación produce. ASM y la proyección real tienen un comportamiento similar, con un error significativamente menor que AAM. Finalmente, usando FaceAPI se obtienen unas

estimaciones con un error muy bajo. Tras la corrección del error sistemático AAM continúa siendo el peor método, con un error en torno a 20 mm. FaceAPI y ASM sufren de un error semejante, inferior a 10 mm. Finalmente, el error utilizando el marcado real es inferior a 5 mm, un resultado realmente bueno (como cabe esperar).

En un estudio más pormenorizado de los errores se observa que en general el error difiere significativamente en los 6 grados de libertad.

En todos los métodos estudiados (salvo el método de referencia que utiliza el propio marcado) el error en *roll* es significativamente menor que en el resto de rotaciones. Este hecho es comprensible, ya que la cara presenta un aspecto altamente simétrico, de modo que es sencillo obtener la rotación de la cara en ese eje, por ejemplo mediante la pendiente de la recta que une los ojos.

Los errores absolutos en *yaw* y *pitch* aumentan notablemente con respecto a roll. Este error se debe principalmente al modelo tridimensional que utiliza cada algoritmo para estimar la información espacial a partir de la imagen plana. Durante la grabación de los vídeos se coloca el sensor sobre la persona en una posición frontal. El problema de este sistema es que la frontalidad es bastante subjetiva dentro de unos límites, de modo que una cara con hasta 5º de pitch puede seguir siendo considerada frontal según el criterio del observador. Por esta razón, dos modelos tridimensionales iguales pero con una rotación de 5º en pitch podrían considerarse frontales. Si esa frontalidad del modelo no es la misma frontalidad de la persona durante la grabación, aparece este error sistemático.

Por ejemplo, dada una imagen de un usuario, supongamos que se estima la rotación mediante dos métodos diferentes, por ejemplo FaceAPI e Intraface. Aunque los dos métodos estimen la rotación de manera totalmente perfecta de acuerdo a sus propios modelos, como utilizan modelos diferentes su resultado será diferente. Por ello, es interesante proporcionar los resultados habiendo eliminado esa componente sistemática del error, ya que de algún modo se elimina ese error propio de cada método. Tras esa corrección del error sistemático, se observa que el error en las 3 rotaciones es semejante.

El mismo problema con el modelo tridimensional se observa en la traslación. En la estimación con los 3 modelos no propios del usuario (el modelo genérico y los dos deformables) se observa que sistemáticamente la estimación de traslación en el eje y es errónea. Lo interesante es que para una misma segmentación mediante ASM o AAM el error con los 3 modelos es muy alto (> 160 mm) pero con el modelo propio el error es muy bajo (< 10 mm). Por ello, el error no está en la segmentación sino en el modelo utilizado. Tras corregir esas discrepancias entre modelos, el error en el eje y disminuye hasta los niveles del error en x (< 10 mm).

Es interesante el error en el eje z, que a pesar de eliminar la componente sistemática sigue siendo alto cuando se usa un modelo diferente al de la persona (> 30 mm en ASM y > 60 mm en AAM). El hecho más relevante es que existe un error en torno a 40 mm al utilizar los *landmarks* reales. Esto indica nuevamente que hay un error proveniente de los modelos que por alguna razón no se elimina con el error sistemático.

Se cree que este error se debe a que se está usando un sistema de visión monocular y el cálculo de la profundidad es una estimación en ocasiones incorrecta. Veamos esto con detalle. Dado un usuario situado frente a la cámara a una distancia z, su proyección en el plano imagen tiene ciertas dimensiones. A partir de esas dimensiones se estima su distancia en base a un modelo tridimensional. Cuando el tamaño de la proyección aumenta se considera que el

usuario se acerca a la cámara, y cuando el tamaño de la proyección disminuye se asume que el usuario se aleja. El problema está en que si se utiliza un modelo sin ninguna información sobre la persona, no se puede asegurar que el tamaño del modelo se corresponda con el tamaño de la cabeza. Por ejemplo, con un modelo de tamaño medio, si la proyección es pequeña se concluye que el usuario está lejos de la cámara, pero podría tratarse de un usuario con la cabeza más pequeña de lo normal (un niño) cerca de la cámara, con el consecuente error en z.

Por ello, si bien por la distancia a la que se han grabado los videos el tamaño del modelo tridimensional no afecta a la estimación de la rotación, la estimación de la traslación puede ser desastrosa si la persona no tiene un tamaño de cabeza comparable al modelo. El problema radica en que con una cámara monocular no se puede conocer el tamaño de la cabeza de la persona, por lo tanto no se puede modificar el tamaño del modelo para ajustarse a ese usuario. Por ejemplo, un objeto grande a cierta distancia de la cámara tiene una proyección semejante a ese mismo objeto con un tamaño menor a menor distancia. En este caso, sin información real sobre el objeto no es posible saber dónde está ni cuál es su tamaño, porque nos encontramos ante 2 incógnitas (distancia y tamaño) y sólo conocemos un dato (tamaño de proyección).

Para solucionar este problema se puede utilizar el modelo marcado con cada usuario. Al tener información de las dimensiones de la cabeza, dada una proyección de esa cabeza sólo puede existir un punto en el que la proyección sea esa. De esta manera la relación proyección – posición es unívoca.

En su defecto, si no se puede obtener el modelo completo de la persona, se puede construir el modelo partiendo de un modelo genérico o deformable y reescalándolo a partir de información conocida del sujeto, como por ejemplo la anchura y altura de la cabeza. La estimación de la rotación será tan buena como con el modelo genérico, pero el error en la traslación se reducirá notablemente.

Tiempo de ejecución

En el tiempo de ejecución se distinguen claramente qué programas son lentos y cuáles están optimizados. Por un lado están AAM y ASM, dos programas implementados en Matlab con las implicaciones que ello conlleva, programas computacionalmente ineficientes pero muy cómodos para ajustar parámetros y mejorar el código. Por el otro lado se encuentran los dos programas comerciales, FaceAPI e Intraface, con una implementación totalmente optimizada para hacerlos tremendamente eficientes, pero que no permiten modificar el código para una necesidad concreta. Estos dos últimos pueden utilizarse en aplicaciones en tiempo real, pero ASM y AAM, en concreto las implementaciones estudiadas en este trabajo, deben utilizarse en aplicaciones *offline*.

Estabilidad

En cuanto a la estabilidad ante escenas estáticas destaca especialmente FaceAPI, con una variabilidad perfectamente despreciable, tanto en lo referente a la posición como a la orientación. Todo lo contrario ocurre con ASM, con una variabilidad superior a 1 px en la segmentación, lo que se traduce en oscilaciones de 0.7º y 2.5 mm. AAM resulta más estable en este aspecto, con una variabilidad en la segmentación inferior a 0.5 px, lo que se traduce en variaciones de 0.2º y 0.9 mm. Muy similar a AAM resulta Intraface, con variaciones de 0.5 px y 0.3º.

11.4 - Ideas finales

En base a los resultados, el primer aspecto relevante a considerar es que el uso de un sistema de visión monocular es una limitación en el cálculo de la posición de la cabeza. Si no se dispone de información sobre el tamaño de la cabeza de la persona, el cálculo de la profundidad o distancia a la cámara es una aproximación que puede resultar desastrosa ante cabezas diferentes al modelo tridimensional.

Por ello, en aplicaciones reales en las que interesa detectar con precisión la posición del sujeto, se necesita más información que la proporcionada por la cámara monocular, bien sea en forma de cámara binocular, o la utilización de referencias de dimensiones conocidas. De no hacerlo, los resultados dependerían mucho del tamaño de la cabeza, algo impensable en una aplicación real (PET, estimulación magnética transcraneal…).

Otro aspecto relevante sobre el uso de modelos tridimensionales es que se debe conocer el sistema de coordenadas sobre el que está referenciado, ya que es una fuente de error sistemático considerable. En una aplicación final sería deseable que el origen estuviera en un lugar fácilmente identificable, como un punto único de la cara (el *corner* de un ojo o el centro de la nariz por ejemplo). En caso contrario, sería deseable realizar algún tipo de calibración que ayude en su localización.

Los algoritmos comparados presentan resultados diferentes en cuanto a la estimación de la posición y orientación de la cabeza. Si bien la traslación hay que tomarla con cautela, los resultados sobre la rotación son claros. ASM + POSIT es el método más preciso, aunque su tiempo de procesado es el mayor de todos. De cualquier modo, para aplicaciones *offline* puede ser un método perfectamente válido. Por ejemplo, en reconstrucción de imágenes tomográficas.

En el caso de disponer de información sobre el sistema, de manera que se pueda corregir el error sistemático, FaceAPI resulta el mejor método, con una variabilidad en la medida inexistente, un tiempo de procesado muy bajo, y un error en la estimación inferior a 1.5º y 10 mm. Este método puede ser utilizado en sistemas prácticamente en tiempo real, como la estimulación magnética transcraneal, aunque con cierta cautela en la traslación.

Sin lugar a dudas, cuanta más información se disponga del usuario mejor será la estimación. Por un lado, utilizar un marcado real de los puntos característicos de la cara, por ejemplo utilizando marcadores fiduciales, reduce el error significativamente, incluso utilizando modelos tridimensionales diferentes a la persona. Por otro lado, utilizar información del sujeto para construir el modelo tridimensional resulta especialmente útil para la estimación de la traslación. Finalmente, si se combinan marcadores fiduciales con un modelo tridimensional real de esos marcadores, el error en la estimación puede resultar prácticamente despreciable, y el tiempo de procesado es irrisorio.

No obstante, el objetivo de este trabajo es comparar la calidad de los algoritmos de estimación de la posición de la cabeza, especialmente los basados en la segmentación de la cara. Recurrir a marcadores fiduciales es a priori un enfoque diferente, y es precisamente lo que se quiere evitar con este trabajo. La investigación sobre los algoritmos que procesan información facial busca prescindir de complejos sistemas colocados sobre el sujeto, que aunque son los más precisos hasta ahora (por eso se utilizan a día de hoy), su diferencia es cada vez menor con los nuevos algoritmos de estimación de la posición de la cabeza actuales, y la comodidad para el paciente cada vez se tiene más en consideración.

12 – Conclusiones

Como punto de partida se ha implementado una base de datos de videos con movimientos 3D de cabeza con movimientos simples y compuestos en los 6 grados de libertad, utilizando una cámara monocular y el sistema de sensores magnéticos trakSTAR.

Se ha automatizado y optimizado el proceso de calibración del sistema cámara-transmisor. El tiempo de calibración se ha reducido de 3 horas a 30 minutos y el error de proyección se ha reducido en un 76%.

Se ha implementado un sistema de marcado automático de las imágenes de la base de datos utilizando dos sensores magnéticos y unas piezas diseñadas para tal fin. El error del marcado automático es de 0.7 mm, y ha evitado un largo proceso de marcado manual.

Se ha mejorado el sistema de grabación, incluyendo funciones de *preview* para la correcta colocación del usuario antes de la grabación, se han automatizado los procesos, y se ha implementado una interfaz gráfica para simplificar la grabación.

Se han comparado 4 algoritmos de estimación de la posición de la cabeza en imágenes monoculares: ASM + POSIT, AAM + POSIT, FaceAPI e Intraface, según los criterios de error en la estimación, tiempo de procesado y estabilidad. También se han estudiado 4 modelos de cabeza: modelo propio de usuario, modelo genérico, modelo deformable por órganos y modelo deformable por PCA.

ASM + POSIT es el método más preciso en la estimación de la rotación, con un error medio de 2.7º. Es a su vez el método más lento en tiempo de procesado (7s/*frame*) y el menos estable ante escenas constantes.

FaceAPI es el método más preciso en la estimación de la rotación tras la eliminación del error sistemático (1.4º). Su tiempo de procesado lo habilita para aplicaciones en tiempo casi real. Es con diferencia el sistema más estable ante escenas constantes.

El cálculo de la traslación presenta serios problemas en cuanto a profundidad al trabajar con imágenes monoculares cuando no se conoce el tamaño real de la persona. El uso de un modelo propio del usuario para estimar su posición resulta exitoso en la resolución del problema.

Tras el modelo propio de cada usuario, el modelo deformable por órganos es con el que se obtienen mejores resultados en la rotación.

Si bien el uso de marcadores fiduciales superficiales en la cara del sujeto puede proporcionar mejores resultados que ASM o FaceAPI, supone una implementación del sistema más compleja, en ocasiones inviable, y es lo que se pretende evitar con los métodos estudiados en este trabajo.

13 – Referencias bibliográficas

[1] Echeverría, Rebeca. Desarrollo de una base de datos de posiciones 3D de la cabeza empleando el sensor trakSTAR 3D GUIDANCE STUDIO

[2] Alexandrov, Pancho. User Guide for the database system with trakSTAR 3D Guidance Studio sensor

[3] Bühler, Paul. An Accurate Method for Correction of Head Movement in PET.

[4] Implementation and Performance of an Optical Motions Tracking System for High Resolution Brain PET Imaging.

[5] Richter, Lars. Fast robotic compensation of spontaneous head motion during Transcranial Magnetic Stimulation (TMS)

[6] Bakker, N.M. Accurate Gaze Sirection Measurements with Free Head Movement for Strabismus Angle Estimation.

[7] Camera Calibration Toolbox for Matlab, http://www.vision.caltech.edu/bouguetj/calib_doc

[8] Roncal, David. Técnicas de Seguimiento de Puntos Faciales y su Efecto en la Estimación de la Posición 3D de la Cabeza

[9] Murphy-Chutorian, Erik. Head Pose Estimation in Computer Vision: A Survey.

[10] Cootes, Tim. An Introduction To Active Shape Models.

[11] DeMenthon, Daniel. Model-Based Object Pose in 25 Lines of Code.

[12] Cootes, Tim. Active Appearance Models.

[13] De la Torre, F. Supervised Descent Method and its Application to Face Alignment

[14] Martins, Pedro. Monocular Head Pose Estimation.

[15] faceAPI, http:// www.seeingmachines.com/product/faceapi/

[16] IntraFace, http://humansensing.cs.cmu.edu/intraface/download.html

[17] Paysan, Pascal. A 3D Face Model for Pose and Illumination Invariant Face Recognition

[18] Moriyama, Tsuyoshi. Meticulously detailed eye region model and its application to analysis of facial images